



## 1. Contexto - Retos de la Hackatón

### **Porcentaje de afinidad: Candidato vs Vacante**

Magneto empleos es una plataforma que permite a personas de todos los sectores laborales encontrar empleos a nivel nacional e internacional y actualmente cuenta con una bolsa de candidatos disponibles para que grandes empresas encuentren los mejores perfiles que se ajusten a sus vacantes.

Esta bolsa de candidatos ha ido creciendo de manera exponencial durante los últimos tiempos. Para cada candidato recopilamos desde información personal como: nombres, apellidos, email, teléfonos, entre otros, hasta información relevante para el proceso de reclutamiento como los estudios, habilidades personales, experiencia y perfil laboral. Aunque toda esta información es posible llenarla por cada candidato, es permitido ingresar en Magneto Empleos candidatos con tan solo el nombre y el correo electrónico y de manera fácil importarlos a una vacante específica.

Esto lleva a que tanto los candidatos con información acorde a la vacante, como candidatos con la información incompleta y los candidatos con información poco relevante para la vacante en particular, se vean de la misma manera y mezclados entre sí, lo que hace que la búsqueda de perfiles que se ajusten con la vacante sean una tarea larga, dispendiosa y costosa.

### **Predicción de la TRM en procesos**

Por otro lado, la Tasa Representativa del Mercado (TRM), es el promedio ponderado (por monto) de las operaciones de compra y venta de dólares en Colombia. Como su nombre lo indica, intenta servir como la cotización representativa del dólar. La TRM tiene importantes usos, como el de servir como precio del dólar para las transacciones bancarias hechas en moneda extranjera y para las anotaciones contables realizadas en moneda extranjera, así mismo, la TRM sirve como principal indicador del precio del dólar en la economía colombiana. La incertidumbre respecto al dólar podría tener impactos negativos en empresas exportadoras e importadoras, por lo que si se contara con una buena predicción, facilitaría la toma de decisiones en el negocio.

## Probabilidad de rotación de empleados

En el proceso de selección de los candidatos para cubrir una vacante, se realizan pruebas psicométricas donde se miden diferentes escalas que predicen la conducta laboral de la persona, aparte de los conocimientos técnicos.

Sin embargo, esta información sólo se utiliza en el proceso de selección y no se tiene en cuenta ya cuando el candidato es contratado y es parte de la empresa.

Sería deseable saber desde la fase de selección, dados los resultados de la pruebas aplicadas, qué tan probable es que un candidato rote en los próximos 3 meses.

## 2. Reto

Este reto se encuentra dividido en dos secciones:

### 2.1 Modelo que permita calcular el porcentaje de afinidad

La primera parte del reto será la de construir un modelo que permita calcular el porcentaje de afinidad (probabilidad de que un candidato avance de la etapa de postulantes o atracción digital en caso que dicha vacante tenga esta etapa) de un candidato en una vacante.

Es importante que el porcentaje de afinidad se ajuste en el tiempo de acuerdo con la información de nuevas vacantes y de nuevos candidatos. Es decir, cada vez que se cree una nueva vacante y lleguen nuevos candidatos a dicha vacante, el modelo debe poder predecir, de acuerdo con la información de la vacante y la información de dicho candidato, qué tan probable es que este candidato avance de la etapa de postulantes o la etapa de atracción digital en caso que la vacante cuente con esta etapa.

Los candidatos (**Candidate**) pueden tener el resultado de pruebas psicométricas (Campo **psy\_tests**). El detalle de estas pruebas se encuentra en el adjunto

#### Consideraciones a tener:

- Las vacantes tienen 2 etapas claves, “**Postulantes**” y “**Atracción Digital**”, las personas que superan estas fases generalmente son candidatos que tienen una afinidad con la vacante.
- Para seleccionar al ganador entre los finalistas, se dará preferencia a los modelos de aprendizaje en línea o de aprendizaje en mini-batches.

### 2.2 Predicción TRM del dólar

La segunda parte del reto consiste en predecir la TRM del dólar en Colombia. El modelo deberá ser capaz de predecir con una periodicidad diaria, la TRM vigente para cada día. Los participantes deberán proveer la data de cualquier feature que decidan incluir en el modelo (por ejemplo, la serie de tiempo de la tasa interbancaria de referencia o el último dato de inflación disponible).

## 2.3 Modelo que permita calcular la probabilidad de rotación de los candidatos

La tercera parte del reto es construir un modelo que calcule la probabilidad de que un candidato rote en la empresa, en menos de 3 meses después de la fecha de ingreso, a partir de los resultados de la pruebas psicométricas.

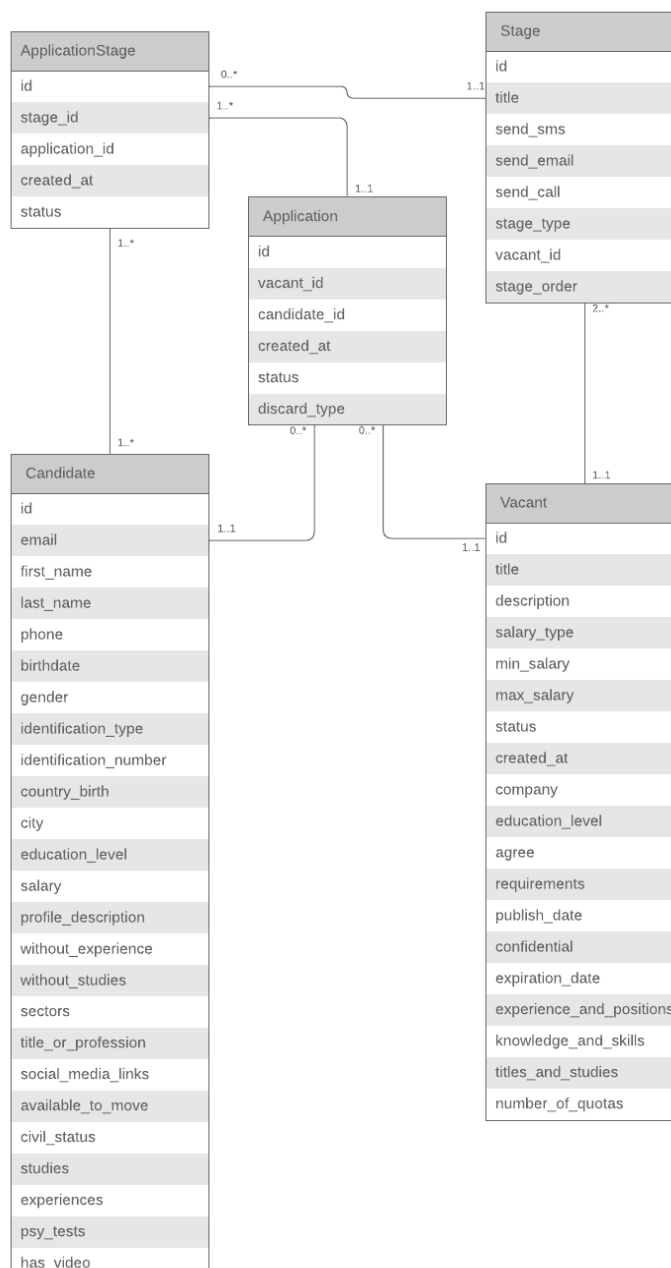
## 3. Insumos y presentación de la data

Para pedir los datasets es necesario enviar firmado y escaneado el **Acuerdo de confidencialidad y Cesión de derechos patrimoniales** ([https://drive.google.com/file/d/1V9OV\\_1IS3WXqcUuYtJsqSHdaWNjUzZHx](https://drive.google.com/file/d/1V9OV_1IS3WXqcUuYtJsqSHdaWNjUzZHx)) por cada miembro del equipo.

El envío es al correo [hackaton@talenta365.com](mailto:hackaton@talenta365.com) con el asunto “**Acuerdo confidencialidad Hackatón 2020**” y debe contener los archivos de todos los miembros del equipo (es decir, solo deben realizar un envío). Se responderá el correo con los links de descarga a los datasets.

### 3.1 Modelo relacional

A continuación se presenta el modelo relacional:



## 3.2 Descripción del modelo relacional - Parte 1 del reto

Dentro de este modelo se tiene que una vacante (**Vacant**) tiene varias etapas (**Stage**), Cada vacante puede relacionarse con varios candidatos (**Candidate**) y esto se ve reflejado en el modelo **Application** donde tiene la relación entre **Vacant** y **Candidate**.

Pero para ver en qué etapa de la vacante está el candidato, se puede realizar por medio del modelo **ApplicationStage**, donde se tiene la relación entre los modelos **Application** y **Stage**. Es decir, cuando un candidato está en una vacante por lo menos tiene que existir un **Application** y un **ApplicationStage**.

La etapa (**Stage**) “Postulante” son las que tienen en el campo **stage\_type** el valor de 0 y la de “Atracción Digital” la que tienen el valor de 1.

## 3.3 Diccionario de datos por dataset - Parte 1 del reto

### 3.3.1 Diccionario de datos dataset Candidate

Nombre del campo	Tipo de dato	Descripción
id	Integer	Entero que identifica un candidato. Es único en todo el dataset de candidatos
email	String	Corresponde al email del candidato. Es único en todo el dataset de candidatos
first_name	String	Nombres del candidato
last_name	String	Apellidos del candidato
phone	String	Teléfono del candidato
birthdate	Date(YYYY-MM-DD)	Fecha de nacimiento del candidato
gender	String	Género del candidato
identification_type	Integer	Tipo de identificación del candidato. Posibles valores  identification: 0, passport: 1, foreigner_id: 2, identity_card: 3, other: 4, CUIL: 5, RUT: 6, CI: 7, RG: 8, CTPS: 9, CPF: 10, PIS: 11, Reservista: 12, PAS: 13, LE_DNI: 14, CARNET_EXT: 15, LM: 16, RUC: 17, P_NAC: 18

identification_number	String	Número de identificación del candidato
country_birth	String	País de nacimiento del candidato
city	String	Ciudad de residencia del candidato
education_level	String	Nivel educativo del candidato
salary	Integer	Aspiración salarial del candidato
profile_description	String	Descripción del perfil del candidato
without_experience	Boolean	Define si el candidato tiene experiencia registrada
without_studies	Boolean	Define si el candidato tiene estudios registrada
title_or_profession	String	Profesión del candidato
available_to_move	Boolean	Disponible para viajar
civil_status	Integer	Estado civil del candidato. Posibles valores:  free_union, separated, divorced, widower, single, married
has_video	String	Video presentación el candidato (campo donde se almacena los videos)
studies	Array<JSON>	Array con objetos tipo JSON con los estudios del candidato, estructura de ejemplo del JSON: <i>{ "id":604, "institute": "UDEA", "title": "Medicina", "start_date": "2018-01-01", "end_date": "2019-01-01", "in_progress": false, "info_studies": null, "study_type": null }</i>
experiences	Array<JSON>	Array con objetos tipo JSON con experiencia del candidato, estructura de ejemplo: <i>{ "company": "Fabrica de licores de antioquia", "position": "Auxiliar de servicio y mercadeo", "start_date": "2018-01-01", "end_date": "2019-01-02", "at_present": false, "description": null, "job_functions": null, "withdrawal_reason": null, "other_withdrawal_reason": null }</i>
psy_tests	Array<JSON>	Array con objetos tipo JSON con pruebas psicológicas del candidato

### 3.3.2 Diccionario de datos dataset Vacant

Nombre del campo	Tipo de dato	Descripción
id	Integer	Entero que identifica a la vacante
title	String	Título de la vacante
description	String	Descripción de la vacante
salary_type	String	Tipo de salario, puede ser alguno de los siguientes valores "Rango salarial" o "Personalizado"
min_salary	Integer	Valor mínimo del rango salarial o valor del salario si el tipo es "Personalizado"
max_salary	Integer	Valor máximo del rango salarial
status	Integer	Estado de la vacante: borrador: 0, publicada: 1, eliminada: 2, expirada: 3, cancelada: 4
created_at	Date(YYYY-MM-DD)	Fecha de creación de la vacante
company	String	Nombre de la empresa
education_level	String	Nivel de estudios mínimo para la vacante
agree	Boolean	Salario a convenir
requirements	String	Requerimientos para aplicar a la vacante
publish_date	Date(YYYY-MM-DD)	Fecha de publicación de la vacante
confidential	Boolean	Marca si la vacante es confidencial
expiration_date	Date(YYYY-MM-DD)	Fecha de expiración de la vacante

experience_and_positions	String	Experiencias o cargos similares
knowledge_and_skills	String	Conocimientos o habilidades deseadas
titles_and_studies	String	Estudios deseados
number_of_quotas	Integer	Número de cupos para la vacante

### 3.3.3 Diccionario de datos dataset Stage

Nombre del campo	Tipo de dato	Descripción
id	Integer	Entero que identifica un stage. Es único en todo el dataset de vacants
title	String	Título de la vacante
send_sms	Boolean	Si la vacante emite comunicado de sms a los candidatos
send_email	Boolean	Si la vacante emite comunicado de email a los candidatos
send_call	Boolean	Si la vacante emite comunicado de llamada a los candidatos
stage_type	Tipo de etapa	Tipo de etapa de la vacante:  applicants( <b>Postulantes</b> ): 0, video_interview( <b>Atracción digital</b> ): 1, finalists: 2, custom: 3, psy_test: 4, psw_test: 5, multiple_psy: 6, caris: 7
vacant_id	Integer	Entero que identifica la vacante a la que pertenece el stage

stage_order	Integer	Posición en la se muestra la etapa en la vacante
-------------	---------	--

### 3.3.4 Diccionario de datos dataset Application

Nombre del campo	Tipo de dato	Descripción
id	Integer	Entero que identifica un candidato. Es único en todo el dataset de vacants
vacant_id	Integer	Entero que identifica la vacante relacionada en el application
candidate_id	Integer	Entero que identifica el candidato relacionado en el application
created_at	DateTime (YYYY-MM-DD HH:MM:SS)	Fecha en la que se creó el application
status	Integer	Estado del application. Posibles valores:  - active: El candidato se encuentra activo en la vacante, disponible para gestionar - discarded: El candidato fue descartado de la vacante (No continua con el proceso) - accepted: El candidato fue promovido a la etapa siguiente y continúa con el proceso - deleted: El candidato fue eliminado por completo de la vacante y no se puede realizar ningún tipo de gestión
discard_type	String	Cuando el estado del application fue discarded, este campo puede tomar dos valores: Manual: Cuando es descartado de manera manual Automatic: Cuando no cumple



		con los requisitos exigidos por la vacante
--	--	--

### 3.3.4 Diccionario de datos dataset ApplicationStage

Nombre del campo	Tipo de dato	Descripción
id	Integer	Entero que identifica la tabla
application_id	Integer	Entero con el ID del modelo Application
stage_id	Integer	Entero con el ID del modelo Stage
created_at	Date(YYYY-MM-DD)	Fecha de creación
status	Integer	Estados disponibles: - active: El candidato se encuentra activo en la vacante, disponible para gestionar - discarded: El candidato fue descartado de la vacante (No continua con el proceso) - accepted: El candidato fue promovido a la etapa siguiente y continúa con el proceso - deleted: El candidato fue eliminado por completo de la vacante y no se puede realizar ningún tipo de gestión

## 3.4 Data - Parte 2 del reto

Para el segundo reto, los participantes podrán descargar la información de la TRM desde la dirección [Tasa Representativa del Mercado \(TRM - Peso por dólar\)](#). No existe ninguna restricción respecto a la data

## 3.5 Data - Parte 3 del reto

Para el tercer reto se utilizara la misma estructura de datos de los candidatos del reto 1 (3.3.1 Diccionario de datos dataset Candidate) y se adiciona el dataset de resultados de pruebas.

### 3.5.1 Diccionario de datos dataset Psychometrics

Nombre del campo	Tipo de dato	Descripción
------------------	--------------	-------------

candidate_id	Integer	Entero con el ID del candidato / Empleado
Fecha_ingreso	Date(DD/MM/AAAA)	Fecha de ingreso del empleado
Fecha_retiro	Date(DD/MM/AAAA)	Fecha de retiro del empleado
Tipo de contrato	String	Texto con el tipo de contrato del empleado
Clasificación	String	Estado del empleado: "Activo" / "Inactivo"
Causales de Retiro	String	Texto con la causa del retiro
Motivos de Retiros	String	Texto con el motivo del retiro
RESPA	Integer	Entero con el valor del resultado en la escalas
ESCALAA	String	Texto con el estado de la escala: "DP" : "Dentro del perfil", "PS" : "Preguntas de seguimiento", "FP" : "Fuera del perfil"
RESPC	Integer	Entero con el valor del resultado en la escalas
ESCALAC	String	Texto con el estado de la escala: "DP" : "Dentro del perfil", "PS" : "Preguntas de seguimiento", "FP" : "Fuera del perfil"
RESPE	Integer	Entero con el valor del resultado en la escalas
ESCALAE	String	Texto con el estado de la escala: "DP" : "Dentro del perfil", "PS" : "Preguntas de seguimiento", "FP" : "Fuera del perfil"
RESPF	Integer	Entero con el valor del resultado en la escalas
ESCALAF	String	Texto con el estado de la escala: "DP" : "Dentro del perfil", "PS" : "Preguntas de seguimiento", "FP" : "Fuera del perfil"
RESPH	Integer	Entero con el valor del resultado en la escalas
ESCALAH	String	Texto con el estado de la escala: "DP" : "Dentro del perfil", "PS" :

		"Preguntas de seguimiento", "FP" : "Fuera del perfil"
RESPQ	Integer	Entero con el valor del resultado en la escalas
ESCALAQ	String	Texto con el estado de la escala: "DP" : "Dentro del perfil", "PS" : "Preguntas de seguimiento", "FP" : "Fuera del perfil"
RESPS	Integer	Entero con el valor del resultado en la escalas
ESCALAS	String	Texto con el estado de la escala: "DP" : "Dentro del perfil", "PS" : "Preguntas de seguimiento", "FP" : "Fuera del perfil"
ITPC	String	Texto con el estado final de la prueba: "DP" : "Dentro del perfil", "PS" : "Preguntas de seguimiento", "FP" : "Fuera del perfil"

## 4. Restricciones de implementación

### 4.1. Formato de entrada del algoritmo

El algoritmo y modelo implementado del **primer reto** deberá contener una función mediante la cual inicie el procesamiento de los dataset y retorne un string que corresponde al path absoluto del archivo de salida con la estructura definida en el literal 4.2

```
/**
 * Este método es usado para dar inicio al algoritmo.
 * @param vacantsPath:string Este primer parámetro corresponde al path
 * absoluto del dataset de vacantes
 * @param applicationsPath:string Este segundo parámetro corresponde
 * al path absoluto del dataset de applications
 * @return outPath:string Path absoluto del archivo de salida acorde a
 * los criterios definidos en el documento técnico
 */

def predict(vacantsPath, applicationsPath, ..., ....):
    ...
    return outPath

/**
 * Esta función es usado para entrenar el modelo.
 * @param vacantsPath:string Este primer parámetro corresponde al path
```

```

*         absoluto del dataset de vacantes
* @param applicationsPath:string Este segundo parámetro corresponde
*         al path absoluto del dataset de applications
*/
def fit(vacantsPath, applicationsPath, ..., ....):
    ...

```

Además de esto, el algoritmo deberá contener un método que permita realizar el training del modelo, es decir, que pueda ajustar el modelo con nuevos datasets y nueva data que surja durante el tiempo.

Para el **segundo reto** se requerirá una función que reciba un archivo con las fechas de los días para los que se desea realizar la predicción de la TRM, tal y como se muestra a continuación:

```

/**
* Este método es usado para dar inicio al algoritmo.
* @param datesPath:string Este primer parámetro corresponde al path
* absoluto del dataset de fechas para las que se desea predecir la TRM
**/

def predict_trm(datesPath):
    ...
    return outPath

```

Para el **tercer reto** se requerirá una función que reciba dos rutas de archivos, con el path del archivo de candidatos y el path del archivo de resultados de pruebas:

```

/**
* Este método es usado para dar inicio al algoritmo.
* @param candidatesPath:string Este primer parámetro corresponde al
path
*         absoluto del dataset de candidatos
* @param psytestPath:string Este segundo parámetro corresponde
*         al path absoluto del dataset de applications
* @return outPath:string Path absoluto del archivo de salida acorde a
*         los criterios definidos en el documento técnico
*/

def predict(candidatesPath, psytestPath):
    ...
    return outPath

```

## 4.2. Formato de salida

Para la salida del algoritmo, se espera que la función **predict** retorne el path absoluto de un archivo que contenga el siguiente formato:

Id_application	Porcentaje_afinidad	Supera_postulantes_y_atraccion_digital
<b>Integer:</b> tomado del dataset applications	<b>Float</b> Se espera que este valor sea entre 0 y 1 incluyendo ambos valores. Ej. 0.654 para 65.4%.	<b>Integer:</b> Se espera que este valor sea 0 o 1 donde 1 corresponde a que sí es probable que supere la etapa de postulantes y atracción digital.

Cada fila deberá contener 3 columnas separadas por coma (CSV), donde la primera columna corresponde al ID del application que se evalúa, la segunda corresponde al Porcentaje de afinidad con la vacante, y el tercero corresponde a la calificación de si es probable o no que un candidato supere las etapas de postulantes y atracción digital.

Ejemplo de salida del primer reto:

```
420,0.56352,0
1230,0.9015,1
4320,0.29325,0
2123,0.3012,0
542,0.7005,0
```

Para el segundo reto, se espera que la función **predict\_trm** tome un archivo con fechas y retorne una predicción de la TRM para cada una de ellas.

Ejemplo de salida del segundo reto:

```
29/02/2020,3535
02/03/2020,3490
03/03/2020,3470
04/03/2020,3485
05/03/2020,3510
```

Ejemplo de salida del tercer reto:

Para la salida del algoritmo, se espera que la función **predict** retorne el path absoluto de un archivo que contenga el siguiente formato:

Id_candidate	Porcentaje_rotacion	Candidato_rotara
--------------	---------------------	------------------

<b>Integer:</b> tomado del dataset candidate	<b>Float</b> Se espera que este valor sea entre 0 y 1 incluyendo ambos valores. Ej. 0.654 para 65.4%.	<b>Integer:</b> Se espera que este valor sea 0 o 1 donde 1 corresponde a que sí es probable que el candidato rote en la empresa en menos de 3 meses
--	---	---

```
678,0.1235,0
222,0.2345,0
123,0.7563,1
234,0.3012,0
345,0.7005,1
```

### 4.3. Restricciones de herramientas y tecnológicas

Para la realización del reto deberá tenerse en cuenta lo siguiente:

- No se permite utilizar servicios externos de procesamiento ni servicios con licencia de pago, es decir, todas las librerías usadas tendrán que ser de carácter público.
- Todo el despliegue deberá ser local y deberá ser implementado con librerías de uso libre sin ningún costo.
- Podrán usar cualquier lenguaje de programación para la realización del reto.

## 5. Criterios de evaluación

### 5.1 Explicación del método de evaluación

#### 5.1.1 Evaluación para el reto 1

**Este punto tendrá un peso del 60% en la calificación total del reto.**

La evaluación del reto se llevará a cabo con un dataset con la misma estructura al entregado, **el cual será entregado una semana antes a la finalización del reto**, en el que se conocen los candidatos y hasta qué etapa llegaron. Se generarán predicciones del modelo usando este dataset y se comparará la salida del algoritmo con la etiqueta real. El dato que se utilizará para comparar los resultados con la realidad de los procesos de selección evaluados, es la tercera columna en el formato de salida, es decir, ***Supera\_postulantes\_y\_atraccion\_digital***.

Esta comparación se realizará utilizando el método del Área Bajo la Curva ROC (AUC por sus siglas en inglés, [https://es.wikipedia.org/wiki/Curva\\_ROC](https://es.wikipedia.org/wiki/Curva_ROC)), donde el resultado es un valor entre 0 y 1.

A continuación se muestra una explicación teórica del significado los valores.

[0.5]: Es como lanzar una moneda balanceada.

[0.5, 0.6): Resultado malo.  
[0.6, 0.75): Resultado regular.  
[0.75, 0.9): Resultado bueno.  
[0.9, 0.97): Resultado muy bueno.  
[0.97, 1): Resultado excelente.

**Los mejores modelos estarán dados por el mayor valor AUC.**

### 5.1.2 Evaluación para el reto 2

**Este punto tendrá un peso del 15% en la calificación total del reto.**

Para la calificación del punto 2, se tendrán como finalistas a aquellos equipos cuyo modelo produzca la más baja RMSE en las predicciones de la TRM durante los 30 días posteriores a la finalización del reto 1. Es decir, se deberá predecir la TRM desde el día 21 de marzo hasta el 21 de abril del presente año.

### 5.1.3 Evaluación para el reto 3

**Este punto tendrá un peso del 25% en la calificación total del reto.**

Al igual que el reto 1, para el reto 3 se utilizara el método del Área Bajo la Curva ROC (AUC por sus siglas en inglés, [https://es.wikipedia.org/wiki/Curva\\_ROC](https://es.wikipedia.org/wiki/Curva_ROC)), donde se compara la data resultado del método implementado vs la data real, tomando la tercera columna de salida **Candidato\_rotara**.

También faltando una semana antes de que se acabe la hackatón se compartirá el dataset de los candidatos y pruebas psicométricas para que realicen la evaluación del método y generen el archivo de salida que será evaluado.

## 5.2 En caso de empate:

Si se llegase a dar un empate entre los concursantes, los criterios de desempate serán:

- **Primer criterio de desempate:** Tiempo de ejecución del algoritmo.
- **Segundo criterio de desempate:** Costo computacional: Recursos utilizados por el algoritmo (Cantidad de memoria RAM).

## 5.3 Jurado:

- **Mario Villamizar Cano:** Líder de Soluciones Digitales y Arquitecto de Soluciones Cloud, Más de 10 años liderando proyectos de investigación, implementando soluciones digitales modernas

(aplicaciones web, móviles, APIs e integraciones), y utilizando prácticas para automatizar y agilizar los despliegues de aplicaciones sobre AWS y Heroku. Es profesor de cátedra del programa de Maestría en Ingeniería de Software en la Universidad de los Andes y co-fundador de Ticketcode, una de las plataformas de gestión de eventos corporativos líder en América Latina. Apasionado por adaptar metodologías, tecnologías y herramientas de Startups al entorno corporativo.

- **Freddy Sánchez Paz:** Arquitecto de software de Talenta365, más de 15 años de experiencia en desarrollo y más de 5 como arquitecto en aplicaciones de misión crítica.
- **Esteban Arango Sánchez:** Líder técnico de Talenta365, Ingeniero de sistemas e informática, Magister en sistemas con más de 10 años de experiencia en desarrollo de software.
- **Danny Pineda Echeverri:** Desarrollador de Python, Data Engineering Analyst.

## 6. Medio de entrega

Cada equipo deberá entregar un ZIP para cada una de las partes del reto, con la siguiente información:

- **Modelo y algoritmo:** Corresponde al algoritmo implementado y los sources necesarios para correr el modelo.
- **Documento de especificaciones:** readme.txt donde se especifique lo siguiente:
  - **Cómo ejecutar el proyecto:** Se debe incluir en el documento un paso a paso de cómo se debe realizar la configuración y el despliegue del proyecto implementado
  - **Descripción del modelo construido:** Se debe dar una descripción de no más de 1000 caracteres en la que se cuente qué estrategia se utilizó para la implementación del modelo.
  - **Herramientas/Librerías utilizadas:** Se debe especificar qué librerías y versiones se utilizaron para la construcción del algoritmo.

Los ZIPs deberán ser enviados al correo: [hackaton@talenta365.com](mailto:hackaton@talenta365.com)

### NOTAS ACLARATORIAS:

1. Todos los entregables deben ser enviados dentro del zip. El no enviar alguno de los entregables será motivo de descalificación. La fecha máxima de entrega será el día 20 de marzo de 2020 a las 23:59:59 hora local.



2. Se podrá descalificar un equipo por copia o fraude no especificado dentro del documento de especificaciones.
3. Los envíos después de la fecha límite no se tendrán en cuenta.