

2022

Talento Tech

TUTORA: NANCY VILLENA
MARIA CARBAJAL

CODER HOUSE CAMADA 34965

Contenidos

Introducción	2
Objetivo	2
Modelo de Negocio	2
Diagramas Entidad – Relación.....	3
Descripción de Tablas.....	3
Vistas	6
Funciones	6
Stored Prodecures.....	7
Script de Creación de objetos	7
Script de Insertar datos	16
Informe.....	19

Introducción

Actualmente existe un gran interés por captar talento de tecnología y estos son muy valorados por lo que se tiene por objetivo formar una base de datos con perfiles completos para elegir a los mejores candidatos. En la empresa donde trabajo se busca apoyar a otras empresas a encontrar el mejor talento tecnológico, por lo que es importante ordenar la información de perfiles en una base de datos.

Objetivo

La empresa “Talento Tech” cuenta con la información de los perfiles de tecnología que proporciona el API de LinkedIn y quisiera explotarlos para obtener métricas en un dashboard para que su equipo de reclutadores pueda tener los insights del mercado como el tiempo promedio de rotación, salario medio para la vacante, entre otras.

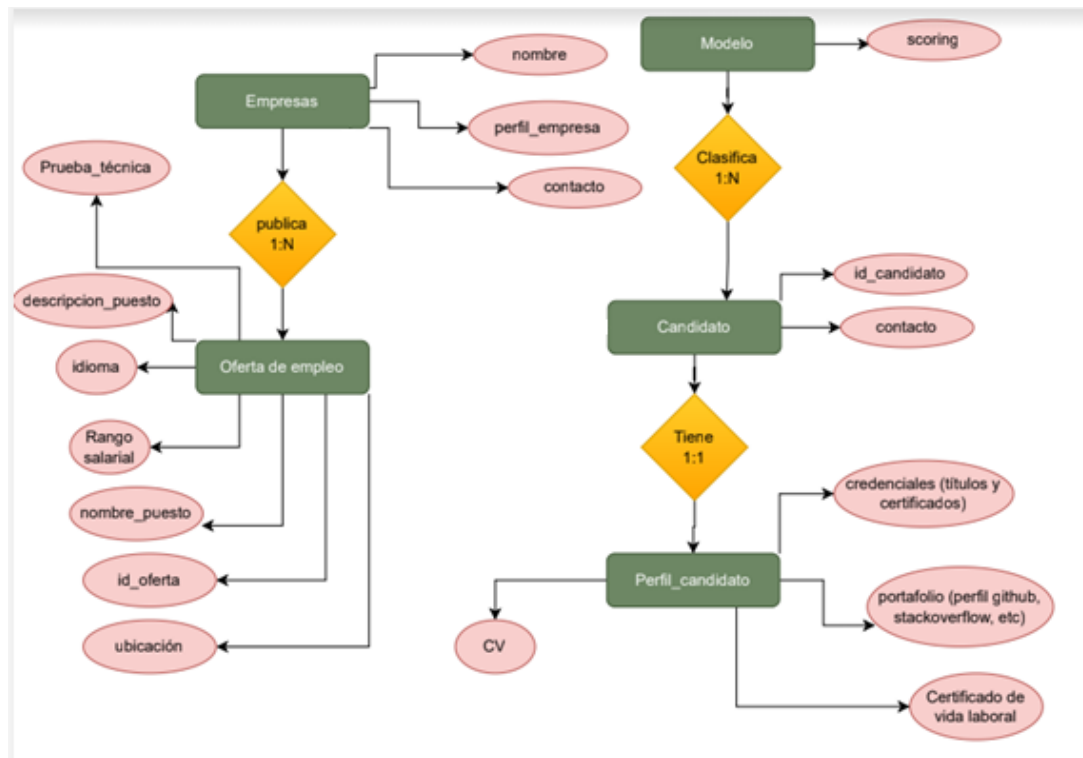
Las empresas tienen el interés de conocer la disponibilidad de talento y los salarios del mercado tanto para retener como para tomar decisiones sobre tecnologías a implementar. Los candidatos pueden validar sus perfiles agregando información haciéndolos más sólidos y atractivos para las empresas, por lo que se les invita a agregar sus redes sociales y perfiles github, stackoverflow, behance, etc.

Modelo de Negocio

El proyecto busca ofrecer datos veraces sobre los candidatos para puestos de tecnología ofreciendo una visión verás de sus perfiles.

- Los candidatos mandan su CV y pueden adjuntar elementos que los validen como sus títulos, certificado de vida laboral de la seguridad social y perfiles de redes sociales.
- Las empresas contactan a los candidatos luego de ver sus perfiles y publican ofertas de empleos.
- La consultora hace un modelo con los datos de los candidatos y crea modelos de scoring para seleccionar a los mejores.

Diagramas Entidad – Relación



Descripción de Tablas

Tabla 1: Empresas

Descripción: En esta tabla están los datos de las empresas que son clientes del servicio y que pueden publicar ofertas de empleo.

KEY	COLUMN	TYPE	LENGHT	NOT NULL	UNIQUE	DEFAULT	COMMENTS
PK	id_empresa	INT	15	NOT NULL	TRUE	AUTO_INCREMENT	Número que identifica a una empresa dada de alta en el sistema
	nombre_empresa	VARCHAR	40				Razón social de la empresa.
	perfil_empresa	VARCHAR	200				Breve descripción de la actividad de la empresa.

Tabla 2: Oferta de empleo

Descripción: En esta tabla se guardan los datos de la oferta de empleo publicadas por las empresas.

KEY	COLUMN	TYPE	LENGHT	NOT NULL	UNIQUE	DEFAULT	COMMENTS
PK	id_oferta	INT	15	NOT NULL	TRUE	AUTO_INCREMENT	Es el número que identifica a la oferta

							publicada por la empresa.
	descripcion_puesto	VARCHAR	4000				Contiene la descripción textual de la oferta de empleo.
	oferta_activa	BOOL		NOT NULL			Es un flag que nos dice si la oferta está aún abierta
	rango_salarial	DECIMAL	6,2				Es el salario anual ofrecido
	idioma	INT	2				Trae el número de idiomas que habla la persona.
	nombre_puesto	INT	80				Nombre del puesto de la oferta
	ubicación	DECIMAL	6,2				Localidad en la que se desempeñaría la actividad laboral.
	experiencia	VARCHAR	20				Nos dice si el candidato debe ser Junior, Senior o un directivo.
FK	id_empresa	INT	15	NOT NULL	TRUE	AUTO_INCREMENT	Número que identifica a una empresa dada de alta en el sistema. Este valor viene de la tabla empresas
	prueba_tecnica	BOOL					Nos indica si para la oferta es necesario someter al candidato a una prueba técnica.

Tabla 3: Modelo

Descripción: En esta tabla se guardan los valores que arroja un modelo de scoring sobre los candidatos.

KEY	COLUMN	TYPE	LENGHT	NOT NULL	UNIQUE	DEFAULT	COMMENTS
PK	id_modelo	INT	15	NOT NULL	TRUE	AUTO_INCREMENT	Número que identifica al modelo ejecutado.
FK	id_candidato	INT	15	NOT NULL	TRUE	AUTO_INCREMENT	Número que identifica al candidato. Este valor viene de la tabla Perfil_Candidato
	scoring	DECIMAL	3,2				Es un valor de propensidad asignado al valor a través de un modelo que ha analizado sus datos.

Tabla 4: Perfil_Candidato

Descripción: La tabla contiene los atributos del perfil de un candidato.

KEY	COLUMN	TYPE	LENGHT	NOT NULL	UNIQUE	DEFAULT	COMMENTS
-----	--------	------	--------	----------	--------	---------	----------

PK	id_perfil_candidato	INT	15	NOT NULL	TRUE	AUTO_INCREMENT	Número que identifica a un candidato que ha ingresado sus datos en nuestro sistema.
	git_hub	BOOL					Nos dice si el candidato ha compartido su perfil de Github en la plataforma.
	stack_overflow	BOOL					Nos dice si el candidato ha compartido su perfil de Stackoverflow en la plataforma.
	vida_laboral	BOOL					Nos indica si el candidato ha dado acceso al certificado de vida laboral de la seguridad social.
	credenciales	BOOL					Esto nos indica si el candidato ha brindado credenciales que prueban sus conocimientos.
	url_git_hub	VARCHAR	50				Guarda la URL del perfil del candidato en Github.
	url_stack_overflow	VARCHAR	50				Guarda la URL del perfil del candidato en Github.
	salario	DECIMAL	6,2				Es el salario que tiene el candidato como expectativa para considerar una oferta.
	experiencia	VARCHAR	20				El candidato se define como Junior, Senior o un directivo.
FK	id_cadidato	INT	15	NOT NULL	TRUE	AUTO_INCREMENT	Número que identifica al candidato. Este valor viene de la tabla Candidato.

Tabla 5: Candidato

Descripción: La tabla datos personales del candidato.

KEY	COLUMN	TYPE	LENGHT	NOT NULL	UNIQUE	DEFAULT	COMMENTS
PK	id_Candidato	INT	15	NOT NULL	TRUE	AUTO_INCREMENT	Número que identifica a un candidato que ha ingresado sus datos en nuestro sistema.
FK	id_perfil_candidato	INT	15	NOT NULL	TRUE	AUTO_INCREMENT	Número que identifica a un candidato que ha ingresado sus datos en nuestro sistema.

	contacto	VARCHAR	6,2				Es el salario que tiene el candidato como expectativa para considerar una oferta.
	url_vida_laboral	VARCHAR	20				La URL al expediente de la vida laboral del candidato.
	url_credenciales	VARCHAR	20				Es una URL que lleva a los documentos de certificados y credenciales presentadas por el candidato.
	CV	VARCHAR	20				Es una URL que lleva al CV del candidato cargado en PDF.

Vistas

Nombre	Descripción	Objetivo
Top Salarios	Muestra los candidatos de la base que actualmente tienen los mejores salarios.	Facilita encontrar los candidatos mejor pagados de la BD.
Top Scoring	Muestra los perfiles que obtuvieron más de 5/10 en las evaluaciones de los modelos.	Da una vista de los candidatos que salieron mejor en las evaluaciones.
Ofertas para expertos	Muestra las ofertas en las que se buscan perfiles con mucha experiencia.	Nos facilita derivar a los consultores más especializados las ofertas que requerirán candidatos expertos, que son difíciles de encontrar.
Promedio Scoring	Muestra el promedio calculado por los modelos de scoring de cada candidato evaluado.	Las evaluaciones son un criterio para elegir al candidato, por lo que tener el promedio nos ayuda a decidir.
Salarios ofertas activas	Muestra los puestos y salarios de las ofertas activas.	Permite hacer un reporting semanal de las ofertas activas al momento.

Funciones

Nombre	Descripción	Objetivo
Calculadora de Salarios	Tomando el id de la oferta, la edad del candidato y el impuesto a la renta (IRPF en entero) se calcula el salario neto que percibiría el entrevistado.	Permite saber al candidato, cuanto recibiría como salario neto, en caso de aceptar la oferta.
Contacto Candidato	Introduciendo el id de oferta, se genera un mensaje para comunicar al candidato, el puesto y empresa donde trabajará	Permite generar un mensaje automático al candidato elegido.

Stored Prodecures

Nombre	Descripción	Objetivo
Control empresas	Registra en la tabla de Audit empresas cuando alguien da de alta a un cliente o lo elimina.	Permite hacer un control de quien da de alta a un cliente o lo elimina en caso toque auditar.
Order ofertas	Ordena las ofertas por id en orden ascendente o descendente.	Permite visualizar las ofertas más recientes y más antiguas para priorizarlas.

Script de Creación de objetos

```
-- Coder House Maria Carbajal
```

```
CREATE SCHEMA `talento_tech` ;
```

```
USE `talento_tech`;
```

```
/*-----TABLAS-----*/
```

```
-- Tabla: EMPRESAS
```

```
CREATE TABLE EMPRESAS (  
    id_empresa int NOT NULL auto_increment,  
    nombre_empresa varchar (40) NOT NULL,  
    perfil_empresa varchar(200) NOT NULL,  
    CONSTRAINT PK_EMPRESAS PRIMARY KEY (id_empresa)  
);
```

```
-- Tabla: OFERTAS
```

```
CREATE TABLE OFERTAS (  
    id_oferta int NOT NULL auto_increment,  
    descripcion_puesto varchar(4000) NOT NULL,
```



```

oferta_activa TINYINT DEFAULT 0,
rango_salarial int NOT NULL,
idioma int NOT NULL,
id_empresa int,
nombre_puesto varchar(80) NOT NULL,
ubicacion int NULL,
experiencia varchar(20) NOT NULL,
CONSTRAINT PK_OFERTAS PRIMARY KEY (id_oferta)
);

-- Tabla: CANDIDATOS
CREATE TABLE CANDIDATOS (
    id_candidato int NOT NULL auto_increment,
    contacto varchar(20) NOT NULL,
    url_vida_laboral varchar(20) NOT NULL,
    url_credenciales varchar(20) NOT NULL,
    cv varchar(20) NOT NULL,
    CONSTRAINT PK_CANDIDATOS PRIMARY KEY (id_candidato)
);

-- Tabla: PERFIL_CANDIDATO
CREATE TABLE PERFIL_CANDIDATO (
    id_perfil_candidato int NOT NULL auto_increment,
    id_candidato int,
    git_hub TINYINT DEFAULT 0,
    stack_overflow TINYINT DEFAULT 0,
    vida_laboral TINYINT DEFAULT 0,
    credenciales TINYINT DEFAULT 0,
    url_git_hub varchar(50) NOT NULL,
    url_stack_overflow varchar(50) NOT NULL,
    salario int NOT NULL,
    experiencia varchar(20) NOT NULL,
    CONSTRAINT PK_PERFIL_CANDIDATO PRIMARY KEY (id_perfil_candidato)
);

```

```
);
```

```
-- Tabla: MODELO
```

```
CREATE TABLE MODELO (  
    id_modelo int NOT NULL auto_increment,  
    scoring int NOT NULL,  
    id_candidato int,  
    CONSTRAINT PK_MODELO PRIMARY KEY (id_modelo)  
);
```

```
/*-----CLAVES FORÁNEAS-----*/
```

```
-- Referencia: FK_OFERTAS (table: OFERTAS)
```

```
ALTER TABLE OFERTAS ADD CONSTRAINT FK_OFERTAS_EMPRESAS FOREIGN KEY  
FK_OFERTAS_EMPRESAS (id_empresa)  
REFERENCES EMPRESAS (id_empresa);
```

```
-- Referencia: FK_PERFIL_CANDIDATO (table: PERFIL_CANDIDATOS)
```

```
ALTER TABLE PERFIL_CANDIDATO ADD CONSTRAINT FK_PERFIL_CANDIDATO_CANDIDATOS  
FOREIGN KEY FK_PERFIL_CANDIDATO_CANDIDATOS (id_candidato)  
REFERENCES CANDIDATOS (id_candidato);
```

```
-- Referencia: FK_MODELO (table: MODELO)
```

```
ALTER TABLE MODELO ADD CONSTRAINT FK_MODELO_CANDIDATOS FOREIGN KEY  
FK_MODELO_CANDIDATOS (id_candidato)  
REFERENCES CANDIDATOS (id_candidato);
```

```
-- End of file.empresas
```

```
/*-----FUNCIONES-----*/
```

```
/*-----Calculadora de salarios-----*/
```

```
/*--Tomando el id de la oferta, la edad del candidato y el impuesto a la renta (irpf en entero)
se calcula el salario neto que percibiría el entrevistado-----*/
```

```
USE `talento_tech`;
```

```
DROP function IF EXISTS `salario_neto`;
```

```
DELIMITER $$
```

```
USE `talento_tech`$$
```

```
CREATE FUNCTION `salario_neto` (num_oferta INT, irpf INT, edad INT)
```

```
RETURNS DECIMAL(8,2)
```

```
reads sql data
```

```
BEGIN
```

```
    DECLARE tax DECIMAL (5,2);
```

```
    DECLARE neto DECIMAL (8,2);
```

```
    DECLARE bono_joven DECIMAL (5,2);
```

```
    SET tax=0.01;
```

```
    SET bono_joven=((72-edad)/18);
```

```
    SET neto = (((SELECT rango_salarial FROM ofertas WHERE id_oferta =
num_oferta)*10000)*((100-irpf+bono_joven)*tax));
```

```
    RETURN neto;
```

```
END$$
```

```
/*-----Generador de mail para candidatos elegidos-----*/
```

```
/*---Introduciendo el id de oferta, se genera un mensaje para comunicar al candidato, el
puesto y empresa donde trabajará-----*/
```

```
USE `talento_tech`;
```

```
DROP function IF EXISTS `contacto_candidato`;
```

```
DELIMITER $$
```

```
USE `talento_tech`$$
```

```
CREATE FUNCTION `contacto_candidato` (remitente varchar(20), mensaje varchar(40),
num_oferta INT, nombre varchar(20))
```

```
RETURNS varchar(8000)
```

```
reads sql data
```

```

BEGIN

    DECLARE puesto varchar(80);

    DECLARE num_empresa int;

    DECLARE empresa varchar(40);

    DECLARE mail varchar(8000);

    SET puesto=(SELECT nombre_puesto FROM ofertas WHERE id_oferta = num_oferta);

    SET num_empresa=(SELECT id_empresa FROM ofertas WHERE id_oferta = num_oferta);

    SET empresa = (SELECT nombre_empresa FROM empresas WHERE id_empresa =
num_empresa);

    SET mail = (concat('Estimad@ ',nombre,': Has sido seleccionad@ para el puesto de ',puesto, '
en ',empresa, '. ',mensaje, '. Un saludo, ',remitente));

    RETURN mail;

END$$

/*-----Stored Procedures-----*/

/*-----Ordena la tabla de ofertas-----*/

/*---Se debe introducir el nombre del campo y ASC o DESC para determinar el ordenamiento---
---*/

DROP PROCEDURE IF EXISTS `sp_order_ofertas`$$

CREATE PROCEDURE `sp_order_ofertas` (IN field VARCHAR(20), IN mi_orden VARCHAR(4))

BEGIN

    /*---Se crean los parametros de ordenamiento del SELECT---*/

    IF field <> '' THEN

        SET @ofertas_order=CONCAT_WS(' ','ORDER BY', field, mi_orden);

    ELSE

        SET @ofertas_order='';

    END IF;

    /*---Se arma la clasula del SELECT---*/

    SET @clausula = CONCAT_WS(' ','SELECT * FROM ofertas', @ofertas_order);

    /*---Seccion de PREPARE y EXECUTE---*/

    PREPARE ejecutar FROM @clausula;

    EXECUTE ejecutar;

    DEALLOCATE PREPARE ejecutar;

```

END\$\$

/*-----Da de alta o de baja empresas-----*/

/*---Para dar de alta se debe introducir el nombre, la descripción y 0 en el id de empresa, para dar de baja se debe insertar el id de la empresa a borrar-----*/

DROP PROCEDURE IF EXISTS `sp_control_empresas`\$\$

CREATE PROCEDURE `sp_control_empresas` (IN accion VARCHAR(20), IN nom_empresa VARCHAR(40) , IN desc_empresa VARCHAR(40), IN id_emp INT)

BEGIN

IF accion='alta' THEN

INSERT INTO empresas VALUES (NULL,nom_empresa,desc_empresa);

ELSE

DELETE FROM empresas WHERE id_empresa=id_emp;

END IF;

END\$\$

DELIMITER ;

/*-----Vistas-----*/

/*-----Top Salarios-----*/

/*----Muestra los candidatos con mejores salarios-----*/

USE talento_tech;

DELIMITER \$\$

CREATE OR REPLACE VIEW highest_salary_view

AS(

SELECT c.*, pf.salario AS 'K eur/año'

FROM candidatos c

JOIN perfil_candidato pf

ON c.id_candidato = pf.id_candidato

group by c.id_candidato

order by pf.salario DESC

LIMIT 5

)\$\$

DELIMITER ;

/*-----Promedio Scoring-----*/

/*----Muestra el promedio calculado por los modelos de scoring de cada candidato evaluado---
--*/

USE talento_tech;

DELIMITER \$\$

CREATE OR REPLACE VIEW avg_scoring_view

AS(

SELECT c.*, AVG(m.scoring)

FROM candidatos c

JOIN modelo m ON c.id_candidato = m.id_candidato

group by c.id_candidato

)\$\$

DELIMITER ;

/*-----Top Scoring-----*/

/*----Muestra los perfiles que obtubieron más de 5/10 en las evaluaciones de los
modelos -----*/

USE talento_tech;

DELIMITER \$\$

CREATE OR REPLACE VIEW top_scoring_view

AS(

SELECT DISTINCT

pf.experiencia AS seniority,

pf.salario AS remuneracion,

pf.id_perfil_candidato AS perfil

FROM perfil_candidato AS pf

JOIN modelo AS m ON (pf.id_candidato = m.id_candidato)

WHERE m.scoring >5

)\$\$

```

DELIMITER ;

/*-----Ofertas para expertos-----*/

/*----Muestra las ofertas en las que se buscan expertos ----*/

USE talento_tech;

DELIMITER $$

CREATE OR REPLACE VIEW experience_view

AS(

    SELECT nombre_puesto,

    experiencia

    FROM ofertas

    WHERE experiencia like '%Expert%'

)$$

```

```

DELIMITER ;

/*-----Salarios ofertas activas-----*/

/*----Muestra los puestos y salarios de las ofertas activas ----*/

USE talento_tech;

DELIMITER $$

CREATE OR REPLACE VIEW salaries_view

AS(

    SELECT nombre_puesto,

    rango_salarial AS 'K eur/año'

    FROM ofertas

    WHERE oferta_activa=1

)$$

```

```

DELIMITER ;

/*-----Triggers-----*/

/*-----Crea Tabla de logs de empresas-----*/

CREATE TABLE audit_empresas (

    id_empresa int PRIMARY KEY,

```

```

    nombre_empresa varchar (40),
    perfil_empresa varchar(200),
    usuario varchar (40),
    hora varchar(200)
);

/*-----Add Empresas-----*/
/*----Registra que se agregó una empresa----*/
CREATE TRIGGER `td_add_new_empresa`
AFTER INSERT ON `empresas`
FOR EACH ROW
INSERT INTO `audit_empresas`
VALUES (NEW.id_empresa, NEW.nombre_empresa, NEW.perfil_empresa,session_user(),
current_timestamp());

/*-----Delete Empresas-----*/
/*----Registra antes de que se borre una empresa----*/
CREATE TRIGGER `td_delete_empresa`
BEFORE DELETE ON `empresas`
FOR EACH ROW
INSERT INTO `audit_empresas`
VALUES (id_empresa, nombre_empresa,perfil_empresa,session_user(), current_timestamp());

/*-----Crea Tabla de logs de ofertas-----*/
CREATE TABLE audit_ofertas (
    id_oferta int PRIMARY KEY,
    id_empresa int,
    nombre_puesto varchar(80),
    usuario varchar (40),
    hora varchar(200)
);

/*-----Add Ofertas-----*/
/*----Registra que se agregó una oferta----*/
CREATE TRIGGER `td_add_new_oferta`

```



```

AFTER INSERT ON `ofertas`
FOR EACH ROW
INSERT INTO `audit_ofertas`
VALUES (NEW.id_oferta, NEW.id_empresa, NEW.nombre_puesto,session_user(),
current_timestamp());

/*-----Delete Ofertas-----*/
/*----Registra antes de que se borre una oferta----*/

CREATE TRIGGER `td_delete_oferta`
BEFORE DELETE ON `ofertas`
FOR EACH ROW
INSERT INTO `audit_ofertas`
VALUES (id_oferta,id_empresa,nombre_puesto,session_user(), current_timestamp());

Repositorio Github: https://github.com/mcarbmont89/coderhouse\_sql

```

Script de Insertar datos

```

/*-----Insertar Datos-----*/

INSERT INTO EMPRESAS (id_empresa, nombre_empresa, perfil_empresa) VALUES
(1,'Coca-Cola','Empresa manufacturadora de bebidas refrescantes'),
(2,'BBVA', 'Institución bancaria española multinacional' ),
(3,'Dragados','Constructora española con presencia en latinoamérica'),
(4,'Klarna', 'Fintech sueca que ofrece microcréditos' ),
(5,'HP','Fabricante de Hardware, equipo de cómputo y servicios tecnológicos'),
(6,'Telefónica', 'Empresa multinacional de telecomunicaciones ' ),
(7,'Sanofi','Laboratorio farmacéutico francés'),
(8,'Bimbo', 'Manufacturadora de productos alimenticios' ),
(9,'Cemex','Cementera mexicana con presencia internacional'),
(10,'Alsea', 'Grupo franquiciario del rubro de la restauración' );

INSERT INTO OFERTAS (id_oferta, descripcion_puesto, oferta_activa, rango_salarial, idioma,
id_empresa, nombre_puesto, ubicacion, experiencia) VALUES

```

(1,'Programador con experiencia en Node.js y desarrollo de APIs',1,50,1,1,'Back-end Developer',40,'Senior'),

(2,'Graduado en Marketing, comunicaciones o afin',1,15,2,2,'Social Media Analyst',40,'Junior'),

(3,'Ingeniero civil o de caminos con experiencia en obras públicas',1,70,1,3,'Director de obras',30,'Expert'),

(4,'Experiencia en cálculo de riesgos financieros para préstamos',1,45,1,4,'Risk Manager',40,'Senior'),

(5,'Vendedor con cartera de empresas medianas',0,40,1,5,'Account Manager',40,'Senior'),

(6,'RF y planificación de radiobases con equipo Huawei y Ericsson',1,35,2,6,'Ingeniero de Radio Frecuencia',40,'Medio'),

(7,'Laboratorista con experiencia en pruebas clínicas',1,32,3,7,'Clinical Trials Manager',40,'Medio'),

(8,'Ejecutivo con amplia experiencia en la industria de manufactura',1,65,1,8,'Director de Operaciones',40,'Expert'),

(9,'Comercial con con experiencia en la industria y cartera de instituciones públicas',0,48,1,9,'Gerente Comercial Gobierno',40,'Medio'),

(10,'Con experiencia en management consulting y finanzas',0,60,1,10,'Director de estrategia',40,'Expert ');

INSERT INTO CANDIDATOS (id_candidato, contacto, url_vida_laboral, url_credenciales, cv)

VALUES

(1,636098478,'/y8567492l','drive.com/mgarcia','in/maugar81'),

(2,633123423,'/n2246263m','dropbox.com/antosua','in/antonellas03'),

(3,637134597,'/f4578357j','icloud.com/elbapaz','in/epaz'),

(4,622345675,'/a3427553s','dropbox.com/ams','in/ams'),

(5,617436874,'/r7537368x','dropbox.com/agarcol','in/anagarcia'),

(6,652657834,'/e4825969w','drive/luccadvitto','in/ladvitto'),

(8,612345733,'/c2395653b','drive/aitorm','in/amentae'),

(9,680467673,'/d2983475c','drive/jruizda','in/javirui'),

(10,632276747,'/w2852852q','drive/rrojas','in/raulrojas');

INSERT INTO PERFIL_CANDIDATO (id_perfil_candidato, id_candidato, git_hub, stack_overflow, vida_laboral, credenciales, url_git_hub, url_stack_overflow, salario, experiencia) VALUES

(1,1,1,1,1,0,'/maugarcia81','/mgarcia',45,'Senior'),

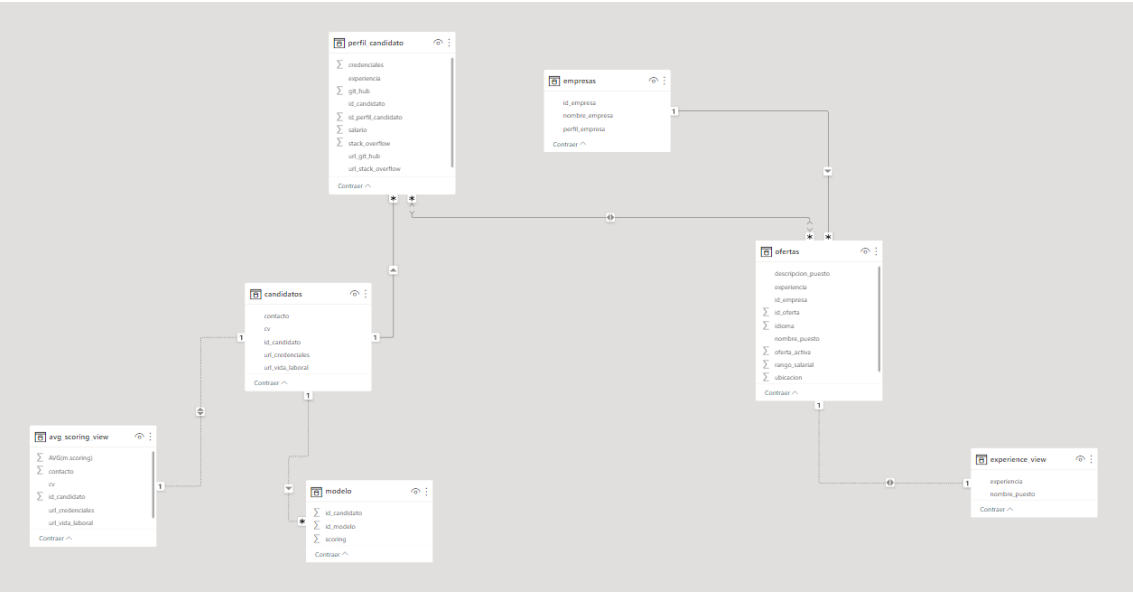
(2,2,0,0,1,0,0,0,12,'Junior'),

```
(3,3,0,0,1,0,0,0,65,'Expert'),  
(4,4,0,0,1,0,0,0,42,'Senior'),  
(5,5,0,0,1,0,0,0,35,'Medio'),  
(6,6,0,0,1,0,0,0,30,'Medio'),  
(8,5,0,0,1,0,0,0,60,'Expert'),  
(9,8,0,0,1,0,0,0,45,'Medio'),  
(10,3,0,0,1,0,0,0,55,'Senior');
```

```
INSERT INTO MODELO (id_modelo, scoring, id_candidato) VALUES
```

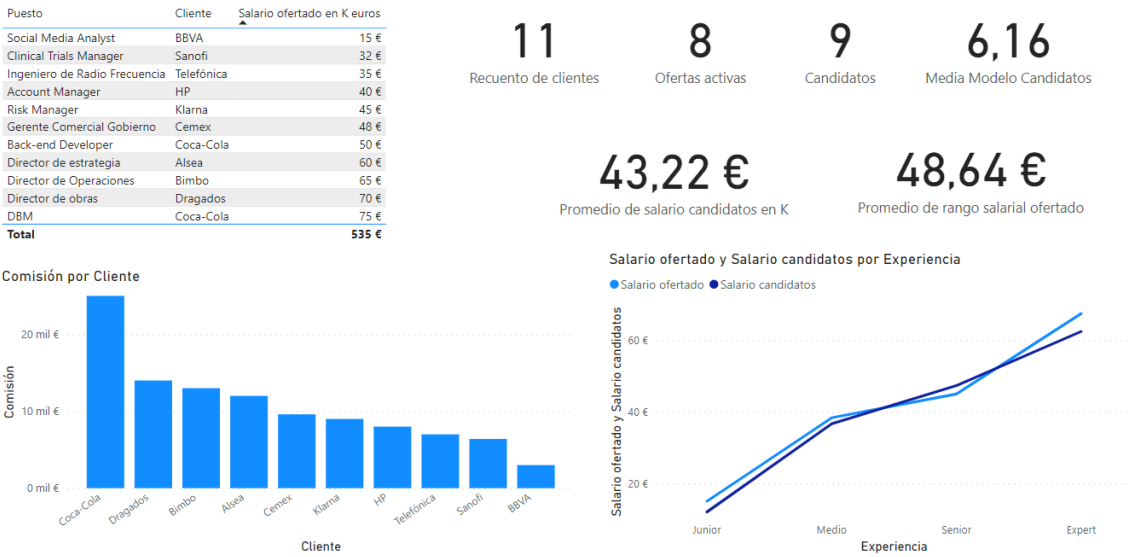
```
(1,7,1),  
(2,5,1),  
(3,9,1),  
(4,5,2),  
(5,7,2),  
(6,3,2),  
(7,5,3),  
(8,8,3),  
(9,3,3);
```

Desarrollo BI



En Power BI fue posible ingestar los datos vía exportación CSV y sobre la plataforma crear métricas y vincular la experiencia de los perfiles con la experiencia solicitada en las ofertas como se puede ver en el esquema.

Informe



En el presente informe se pueden apreciar el listado de puestos con los salarios que se ofrecen, al mismo tiempo que debajo se puede ver la comisión que cada cliente generaría si se logra conseguir al candidato que solicitan. En Power BI fue posible calcular la comisión como el 20% del salario anual ofertado. Por otra parte, para comparar el salario de las ofertas con el de los candidatos, tenemos el gráfico de línea de tendencia. Donde se aprecia que la media de los

salarios para nivel de experiencia junior, medio y expert ofrecidos por los clientes son superiores al de los que actualmente tienen los candidatos y esto denota que hay buenas posibilidades de cerrar dichas posiciones.

Respecto a los indicadores a monitorear contamos con el número de clientes, las ofertas y los candidatos. En este negocio se recomienda tener entre 3 y 10 candidatos para hacer la selección y luego ir descartando hasta cerrar la posición. Además de la tabla “Modelo” se obtiene una media de lo obtenido en test y evaluaciones para indicarnos del 1 al 10 qué tan adecuados son para la posición, de este modo, Operaciones puede tomar acción y buscar más candidatos.

Conclusiones y líneas futuras de trabajo

Respecto a los futuros avances, lo ideal, es tener una interfaz web o gráfica que permita asociar diversos perfiles a las ofertas. Esto es lo que hacen ATsS (applicant tracking system) formando una relación muchos a muchos entre ofertas y perfiles de candidatos.

Al llegar a una masa crítica de candidatos y si exploramos los datos de los perfiles de LinkedIn se pueden hacer estudios que tomen en cuenta la ubicación geográfica de los candidatos para determinar con cuantos perfiles de cierta competencia contamos.

Finalmente, aplicando técnicas de Inteligencia Artificial como NLP se podrían crear a partir del texto de descripción de experiencia laboral, nuevas variables para hacer un scoring más profundo de los candidatos.