

Búsqueda de empresas de interés: información a tener en cuenta

TIPOS DE EMPRESAS

En una Corporativa (Big Company) las funciones del puesto están pautadas y todos los procesos bien definidos - tendremos claridad en cuanto a lo que hay que hacer y escasa toma de decisiones. Será una oportunidad para ser mentorizado y aprender mucho de otros y el recorrido en cuanto a la carrera profesional es bastante más lento. El proceso de selección puede tener hasta 6 o 7 fases. Ejemplo paradigmático de nicho es el Sector Banca.

En una Startup los procesos no están tan bien definidos y puede que te encuentres haciendo cosas que no tienen que ver con la posición y una mentorización más laxa, pero tendrás más autonomía y capacidad de decisión. El proceso de selección es más corto e incluso puede que tengamos una fase de entrevista con el CEO o Fundadores de la propia empresa.

En cuanto a **perfiles**, no es necesario tener un background STEAM. Si bien, las personas que no cuentan con él, ocupan puestos ligados al área negocio, en los que la formación No Técnica, pueda aportar visiones diferentes y de valor para sus clientes internos y externos (serían ADE, Derecho y Psicología). Se recomienda dirigirse al Área de su sector anterior, si vienen de una trayectoria no STEAM, para poder también aportar expertise en ese sector como valor añadido.

PREPARACIÓN DE ENTREVISTAS

LA PRUEBA TÉCNICA

Encontramos prueba técnica en el **99% de las entrevistas**. Normalmente es una prueba técnica en la que te dejan unos días para resolverla en casa pero también puede haber una prueba presencial (live).

Objetivo: lo importante no será tanto obtener resultados sino mostrar nuestra capacidad de análisis y razonamiento frente al reclutador y justificar las decisiones que hemos ido tomando para sacar la prueba adelante. La prueba está enfocada a averiguar si el candidato tiene una buena métrica, y si entienden a qué se traduce esto (negocio).

Contenido: suelen ser casos bastante específicos, relacionados con la propia empresa para ver si somos capaces de entender lo que se está haciendo y escalarlo a negocio. Puede haber conceptos teóricos (ej: *¿Qué es la regresión logística?* / *¿Cómo limpiarías esta BBDD?*) y de Machine Learning y caso de uso de ML, SQL (ej: *Con los datos que te facilitamos, predice la temperatura de mañana o crea un modelo para detectar anomalías en XX*). Hay que tener en cuenta que una tipología de preguntas muy frecuente son problemas de optimización.

Tipos de pruebas:

- **Prueba live.** Prueba en vivo con un pequeño reto y tiempo limitado. Normalmente la haces con dos personas de la empresa.
- **Prueba para hacer en casa.** Tienes que entregar un proyecto o ejercicio. Puede tener tiempo de entrega (días o semanas).
- **Una charla técnica con preguntas teóricas y situaciones posibles.** Para conocer tu nivel.

Esta clasificación se aplica a Full Stack aunque para Data Science también aplica. Nos enfocamos en Live porque la entrevista genérica de DSC se explica detalladamente en otros apartados posteriores.

Cosas a tener en cuenta y saber:

- La gran mayoría de pruebas son muy parecidas - te proporcionan un dataset, te piden que lo analices y construyas un modelo predictivo.
- Saber diseñar y entrenar uno o varios modelos predictivos - eligiendo las técnicas adecuadas de selección y optimización de hiper parámetros.
- Importante - Saber definir y elegir bien el train, validation y test set.

MATERIAL PARA PRUEBA TÉCNICA

Muchas veces las preguntas son muy estándar. Buscando en Google compilaciones de preguntas típicas de data science o data analytics, se pueden encontrar muchos recursos muy válidos. Un buen recurso sería mirar soluciones del famoso dataset de titanic: <https://www.kaggle.com/c/titanic>.

(Libro gratis online) - <https://themlbook.com/> resumen de todo machine learning, y puede ayudarte mucho a preparar la entrevista técnica.

COMPETENCIAS SOFT MÁS IMPORTANTES Y PREGUNTAS PARA DETECTARLAS QUE PUEDEN HACERNOS

- Capacidad crítica y de análisis
- Proactividad y curiosidad
- Trabajo en equipo
- Capacidad de comunicación y presentación (del proyecto al equipo técnico y no técnico)
- Resolución de problemas y creatividad a la hora de buscar diversos enfoques para hallar una solución.

Pregunta 1 (Trabajo en equipo)	<i>Cuéntame alguna vez que tuvieras un conflicto en el trabajo y cómo lo resolviste</i>
Pregunta 2 (Cap. crítica y análisis)	<i>Llevas mucho tiempo y esfuerzo dedicado a un proyecto que no está funcionando, ¿qué haces?</i>
Pregunta 3 (Comunicación)	<i>¿Qué importancia tiene para ti la comunicación en tu puesto de trabajo?</i>
Pregunta 4 (Creatividad y resolución problemas)	<i>¿Me podrías hablar de un proyecto donde te hayas salido del camino y de los procesos habituales? ¿Qué resultados has obtenido?</i>

¿CÓMO DESTACAR EN UNA ENTREVISTA?

Estudiar a conciencia la empresa, sus proyectos -pensando cómo se podrían mejorar y teniendo preguntas sobre ellos preparadas-, el equipo y el motivo de la utilización de datos en la compañía.

Preparar una **carta de presentación** personalizada justificando por qué las capacidades y perfil que necesitan se alinean con tus proyectos y experiencias pasadas.

Preparar la **prueba técnica** previamente haciendo simulacros y creando escenarios y resolviéndolos.


Cuando no tenemos una primera experiencia podemos ofrecer servicios como freelance a través de webs como: *Fiverr, Toptal, Malt, X-team, Upwork, Turing* para añadir a nuestro CV.

¿QUÉ DEBEMOS EVITAR EN UNA ENTREVISTA?

No conocer la empresa y haberse preparado previamente en este sentido y no hacer preguntas finales - imprescindible mostrar interés a través de preguntas y que sea una conversación, no un monólogo del reclutador/a-.

Intentar proponer soluciones a problemas excesivamente difíciles (como una red neuronal) para demostrar nuestras capacidades. Es preferible adherirse a soluciones más fáciles- aunque sean más tradicionales que funcionan igual de bien-.

ENTREVISTA TIPO DATA SCIENCE

 **Importante:** Cuando nos pregunten debemos evitar lanzarnos a dar una respuesta; intentar entender y acotar el problema que te plantean para así ofrecer la solución más adecuada.

-----X-----

A modo introductorio y posterior a una presentación de nuestra trayectoria profesional, se suelen hacer preguntas muy genéricas para ver la capacidad de comunicación del candidato y valorar la adecuación y comprensión del puesto de trabajo. Algunas preguntas de este tipo son:

- Para ti, ¿qué es un Data Scientist? Alguien capaz de utilizar datos para poder tomar decisiones e incidir directamente en los resultados de la empresa.
- ¿Por qué has optado por esta posición y por qué crees que puedes ser la persona indicada para cubrirla? ¿Qué puedes aportar a la empresa? Importante haber estudiado la empresa y posición y saber lo que están buscando para adaptar nuestro mensaje.
- ¿Dónde te ves en 5 años? Importante que, al haber mucha rotación en el sector, tengamos pensada esta respuesta para mostrar que buscamos un proyecto con continuidad.

Una vez superada esta fase de presentación / preguntas generales, se suelen afrontar preguntas sobre los aplicativos que utilizamos, preguntas enfocadas a Soft Skills o también **preguntas generales y básicas pero ya relacionadas con la posición de Data**. Algunas preguntas de este tipo son:

- ¿Qué tecnologías utilizas normalmente para hacer Data Science? Es interesante que sepamos qué tecnologías utilizan en la empresa para adecuarlo y no limitarnos a nombrar las tecnologías, sino explicar para qué utilizaremos una u otra para mejorar en eficiencia y obtención de resultados. Por ejemplo: A nivel de gestión de BBDD sé trabajar con MySQL, PostgreSQL, MongoDB, Dynamo DB. Lenguajes de programación domino Python o R. Machine Learning puedo trabajar con Tensor Flow, Sklearn, etc. Por último, de herramientas de visualización también puedo trabajar con Tableau o Power BI.
- Dame un ejemplo de alguna vez que hayas aportado valor a algún stakeholder a través de datos. El objetivo principal de esta pregunta es detectar la capacidad de influencia y la toma de decisiones. Es importante alinear la ciencia de datos con los objetivos de la empresa.
- ¿Cuál es la diferencia entre un científico de datos y un analista de datos? ¿Dónde puedes encajar mejor? Un científico de datos recolecta, procesa y analiza datos para proporcionar predicciones para el negocio. Sin embargo, un analista de datos resuelve problemas ya existentes y está más centrado en el análisis estadístico. Para valorar dónde podemos encajar mejor nos basaremos en nuestras habilidades, experiencia y la posición a la que postulamos.
- ¿Cómo determinarías si un conjunto de datos es de buena calidad? Se trata de una pregunta teórica para conocer si la persona tiene conocimientos básicos. Cada uno puede aportar la respuesta que considere oportuna pero una buena respuesta sería que la BBDD no tenga sesgos ni alteraciones y que los datos sean exactos, precisos, íntegros, singulares y legibles. A partir de aquí, podemos centrarnos en cómo conseguir pasar de una BBDD de mala calidad a una de buena calidad

La siguiente fase de la entrevista se vuelve un poco más complicada. Suelen mezclar **preguntas de conocimiento (parecido a un examen) con preguntas para valorar nuestras soft skills**:

- ¿Cuáles son las tareas que harías o te gustaría hacer como data scientist? Se trata de una pregunta utilizada para conocer el perfil de trabajador.
- Háblame sobre algún proyecto en el que te hayas enfrentado a un gran problema. ¿Cómo lo manejaste? Debemos centrarnos en algún problema profesional y, si puede ser, relacionado con datos o con el manejo de alguna BBDD. La respuesta correcta consiste en nombrar el problema pero centrarse en la solución y tu papel en la resolución del mismo.
- ¿Cuál es la mayor BBDD que has manejado, cómo la procesaste y cuáles fueron los resultados?
- ¿Qué es el sobreajuste de datos y cómo lo reducirías? El sobreajuste se ve cuando un modelo no está bien generalizado y capta ruido, se puede reducir recogiendo más datos o buscando modelos más sencillos, por ejemplo.
- ¿Qué pasos seguirías para un proyecto de análisis? Se trata de analizar un proyecto mediante datos. Para ello, una buena pauta para realizarlas es:
 1. Definir el problema
 2. Explorar datos
 3. Preparar datos
 4. Modelar datos
 5. Validar datos
 6. Seguimiento
- ¿Qué es la regresión logística? Se trata de un método descriptivo basado en estadística utilizado para revisar un conjunto de datos donde hay una o más variables independientes que definen un resultado.
- ¿Qué técnicas de machine learning has implementado? Árboles de decisión, clasificadores lineales, support-vector machine, redes neuronales básicas.... La clave está en explicar aquellas que hayamos tocado a nivel académico o profesional, explicar en qué consiste, su utilidad, funcionamiento y resultados que arroja.
- ¿Cómo has evitado que tus modelos arrojen sesgos? Se puede hablar del tipo de sesgos (muestreo, exclusión, prejuicio...) y también de cómo los has evitado.

Otras preguntas técnicas para la preparación de la entrevista:

- **¿Qué haces si tienes un dataset con muchas columnas? Pregunta clásica (ordenadas de peor a mejor respuesta)**

o Preguntas a realizar:

§ ¿De cuántas columnas estamos hablando? 500, 1.000, 100.000

§ ¿El modelo debe ser entendible por negocio? Si es que sí, queremos utilizar variables originales no transformaciones o combinaciones de estas.

o Seleccionar atributos que tengan sentido para negocio y en los que negocio pueda actuar. A menudo el modelo más preciso no es el mejor.

o Deshacerte de atributos con alta correlación entre ellos. ¿Pero y si aún seguimos con muchas columnas?

o Realizar permutaciones con las columnas, lanzar un modelo por cada permutación y ver qué variables son más predictoras. (Time consuming)

o Utilizar un transformer y quedarte con el output del encoder. (no respondería directamente con esta respuesta, pero la mencionaría como una posibilidad para desmarcarme)

* Un transformer es una red neuronal que transforma tu input en un output con menos variables (las que tú elijas). El output reducido se introduce como input a una nueva red neuronal que tiene como output el input original. De esta manera expresas tu tabla de muchas dimensiones a menos dimensiones. La red neuronal expresa el output como una combinación lineal de las variables del input.*

- **A/B Testing**

- o Preguntas a realizar.

- § ¿Estamos comparando medias o desviaciones? ¿La distribución a observar podría considerarse normal?

- o Z-test (media y distribución normal), t-test (media y distribución normal, si tienes pocas muestras), chi2 (media y dist. categorizada), F-snedecor (desviación)

- o Entender que es el p-value (valor que hace que el test sea significativo o no)

- **¿Qué es p-value?**

- o El p-value permite es la probabilidad para poder rechazar o no el test de hipótesis realizado, es decir, para ver si los resultados son significantes o no.

- **¿Qué es un test de hipótesis?**

- o Un test de hipótesis consiste en demostrar que la diferencia en los datos (test y control) no es debida a factores random. Si es random significa que no existe diferencia entre test y control.

- o Rechazamos la hipótesis, si los datos confirman que test y control son distintos. Y la confirmamos si decimos que pertenecen al mismo grupo.

- **Underfitting y overfitting**

- o ¿Qué es? Saber su definición.

- § El underfitting ocurre cuando el modelo es tan simple que no es capaz de detectar y captar los patrones de los datos (los resultados en train test son igual de malos).

- § El overfitting es cuando el modelo ha memorizado los datos del train y no es bueno generalizando. Los resultados en train son buenos pero en test falla. El modelo tiende al Variance.

- § Saber qué es Bias y Variance.

- **¿Qué haces para combatir el overfitting?**

- o Regularizar el modelo (quitando profundidad, complejidad, reducir hyperparameters que generan overfitting).
- o Reducir el número de columnas que se entran al modelo.
- o Utilizar métodos de sampleo o validación. K-fold cross-validation

- **¿Qué haces para combatir el underfitting?**

- o Aumentar la complejidad del modelo. O bien cambiar de tipo de modelo (pasar de regresión lineal a Random Forest por ejemplo). Modificar los hyperparameters.
- o Aumentar la cantidad de datos. Recopilar más y mejores datos.

- **Muestra desbalanceada**

- o Clasificación binaria, aunque también habría que pensar qué pasaría con una multclasificación (respuestas ordenadas de peor a mejor)
 - § Balancear la muestra. ¿Cuándo harías undersampling y oversampling? Si tenemos pocas muestras, oversampling mejor. Si tenemos muchas, mejor undersampling.
 - § Generar datos sintéticos con SMOTE. Este enfoque suena muy bonito, pero hay muchos data scientist en contra de utilizarlo.
 - § En caso de imágenes, data augmentation funciona genial.
 - § Modificar la función de coste, de manera que penalice más error en la categoría con menos muestras. O bien tocar los pesos de la selección de las muestras o la ponderación del error de cada categoría.
- o Regresión Lineal (aquí jugamos con la métrica)
 - § Mínimos cuadrados: RMSE (clásica, le afectan mucho los outliers. Sólo recomendada para distribución normal)
 - § MAE (no le afectan los outliers pero tarda más en entrenar el modelo)
 - § Log Error: RMSLE (valores que suelen tener un rango de valores comunes, queremos detectar cuando obtenemos este valor común = Precios de casas. La mayoría de casas suelen estar entre los 50-100 mil, pero habrá casas de millones)
 - § Tweedy (valores que suelen tener un valor común, queremos detectar cuando NO obtenemos este valor común. Ventas diarias de un producto en una tienda = por ejemplo Zapatos de una marca en una tienda de un pueblo, la mayoría de días tenemos 0 ventas).

- **Preguntas de matriz de confusión. ¿Qué es peor: error de tipo I o tipo II?**

- o Error de tipo I: Falso Positivo "Un inocente va a la cárcel". Ejemplo: Se declara que un paciente tiene una enfermedad mortal cuando en realidad hubo un fallo en el test.
- o Error de tipo II: Falso Negativo "Un criminal queda suelto". Ejemplo: En el escáner del aeropuerto una persona lleva una navaja y el sistema no pita para no incomodarle.

- **¿Qué haces con una columna con muchas categorías? (de peor a mejor)**

- o No considerar esta columna por su cardinalidad.
- o Reducir estas categorías a grupos más generalizados si es posible. España, Francia => Europa; EEUU, Canadá => América
- o Hacer mean encoding. ¿Qué valor de media respecto a la variable objetivo tiene cada categoría?
- o Quedarte con las 10 categorías más predominantes y aquellas que no lo sean codificarlas como Otros.
- o EXTRA. Si hay pocas categorías => Se puede realizar One Hot Encoding. Recordar que OHE no es necesario en árboles de decisión y a veces puede ser contraproducente.

- **¿Qué haces con nulos?**

- o Pregunta clave: ¿Estamos tratando con registros ordenados en el tiempo Time Series o no?
 - § No Time Series (de peor a mejor)
 - Imputar con valor constante: -1, media, mediana
 - Imputar utilizando KNN. Muy costoso computacionalmente
 - Imputar con la mediana de la categoría a la que pertenece (rellenar sueldos nulos de clientes de España con el sueldo mediano de España)
 - Anotar en una columna extra que registros se ha imputado el nulo (puede ser un indicador para el modelo, esta respuesta te hará desmarcarte)
 - § Time Series (NUNCA rellenar con valor constante, sino destrozas la tendencia temporal. El objetivo es intentar seguir con la tendencia)
 - Fill forward. Rellenamos con el dato anterior hacia delante.
 - Fill backward. Rellenamos con el dato posterior hacia detrás.
 - Media móvil de 3 forward. Cogemos la media de los 3 valores anteriores.
 - Media móvil de 3 backward. Cogemos la media de los 3 valores posteriores.

- **¿Qué es el ROC AUC score? ¿Accuracy? ¿Precisión? ¿Recall? Cuando importa más recall que precisión**

- o ROC AUC score es el área que hay al graficar el True Positive Rate (TPR) y el False Positive Rate (FPR) para distintos thresholds. El TPR indica que tan bueno es el modelo reconociendo los casos positivos (también llamado como recall). El FPR indica que tan bueno es el modelo reconociendo los casos negativos.
- o La accuracy son todos los aciertos que hace nuestro modelo respecto a todos los casos.
- o Precisión indica lo acertado que es nuestro modelo con los casos positivos. Queremos pocos casos donde erremos.
- o Recall indica que tan bueno es el modelo reconociendo los casos positivos, pero no significa que vaya a acertar. Estamos dispuestos a errar, pero queremos reconocer todos los casos positivos.
- o Precision y recall son contrapuestas, si queremos reconocer muchos de los casos positivos, el modelo podría decir que todos los casos son positivos; pero acertaría poco dado que muchos casos serían error.

o Depende... Si queremos reducir false positive precisión. Si queremos reducir los False Negative recall.

- **¿Qué haces con Outliers? (de peor a mejor)**

o Preguntas a realizar:

§ ¿Qué considerarías un outlier? Un registro con valores erróneos o con valores extremos. ¿Con qué variable estamos tratando?

§ ¿Tiene sentido que tenga valores negativos? Precios de Venta no deberían ser negativos

§ ¿Cuánto % de los datos son outliers?

§ Tenemos outliers en el input (en los datos) o en el output (en el target)

o Aquí diferenciaría entre cómo detectar los outliers y cómo actuar:

§ Detectar outliers

- Valores fuera de los 05 y 95 cuantiles
- Valores extremos tras normalizar con Z-score
- Valores sin sentido

§ Actuación frente a outliers

- Eliminar los registros con outliers (Cuidado! Nos podemos permitir el lujo de dejar de considerar estos registros. ¿Cuántos registros nos quedan?)
- No hacer nada y utilizar modelos a partir de árboles de decisión a los cuales no les afectan mucho los outliers.
- Imponer un máximo a los outliers, de manera que la columna no exceda el máximo impuesto (clip en python)

- **¿Qué diferencias hay entre Random Forest y Boosting?**

o Random Forest es un ensemble de varios árboles de decisión, donde cada árbol selecciona una muestra o sample del dataset. La solución final es la media de muchos árboles entrenados con diferentes muestras del dataset.

o Boosting (XGBoost o LightGBM) es un ensemble de árboles de decisión simples. El primer árbol se entrena con todos los datos, el siguiente modelo tiene como objetivo reducir el error del modelo anterior y el siguiente modelo corregir los errores del anterior... así sucesivamente. Esto puede hacerse dando pesos a los registros (de manera que penalizamos más los casos donde más hemos errado = AdaBoost) o trabajar con los errores directamente (Boosting). La solución final es la media de las estimaciones de todos los árboles de decisión.