



GUÍA DE EVIDENCIAS DE LA UNIDAD DE COMPETENCIA

“UC2497_3: Desarrollar componentes software específicos para sistemas de inteligencia artificial basados en aprendizaje automático”

CUALIFICACIÓN PROFESIONAL: GESTIÓN DE LA INSTALACIÓN, DESPLIEGUE Y EXPLOTACIÓN DE SISTEMAS DE INTELIGENCIA ARTIFICIAL BASADOS EN APRENDIZAJE AUTOMÁTICO.

Código: IFC750_3

NIVEL: 3

1. ESPECIFICACIONES DE EVALUACIÓN DE LA UNIDAD DE COMPETENCIA.

Dado que la evaluación de la competencia profesional se basa en la recopilación de pruebas o evidencias de competencia generadas por cada persona candidata, el referente a considerar para la valoración de estas evidencias de competencia (siempre que éstas no se obtengan por observación del desempeño en el puesto de trabajo) es el indicado en los apartados 1.1 y 1.2 de esta GEC, referente que explicita la competencia recogida en las realizaciones profesionales y criterios de realización de la UC2497_3: Desarrollar componentes software específicos para sistemas de inteligencia artificial basados en aprendizaje automático.

1.1. Especificaciones de evaluación relacionadas con las dimensiones de la competencia profesional.

Las especificaciones recogidas en la GEC deben ser tenidas en cuenta por el asesor o asesora para el contraste y mejora del historial formativo de la persona candidata (especificaciones sobre el saber) e historial profesional (especificaciones sobre el saber hacer y saber estar).

Lo explicitado por la persona candidata durante el asesoramiento deberá ser contrastado por el evaluador o evaluadora, empleando para ello el referente de evaluación (UC y los criterios fijados en la correspondiente GEC) y el método que la Comisión de Evaluación determine. Estos métodos pueden ser, entre otros, la observación de la persona candidata en el puesto de trabajo, entrevistas profesionales, pruebas objetivas u otros. En el punto 2.1 de esta Guía se hace referencia a los mismos.

Este apartado comprende las especificaciones del “saber” y el “saber hacer”, que configuran las “competencias técnicas”, así como el “saber estar”, que comprende las “competencias sociales”.

a) Especificaciones relacionadas con el “saber hacer”.

La persona candidata demostrará el dominio práctico relacionado con las actividades profesionales que intervienen en Desarrollar componentes software específicos para sistemas de Inteligencia Artificial basados en aprendizaje automático, y que se indican a continuación:

Nota: A un dígito se indican las actividades profesionales expresadas en las realizaciones profesionales de la unidad de competencia, y a dos dígitos las reflejadas en los criterios de realización.

1. Modelizar algoritmos de carácter general y específicos de Inteligencia Artificial, previa interpretación, usando lenguajes propios del sector para integrarlos en sistemas de Inteligencia Artificial.

- 1.1 Los problemas a resolver se interpretan, abstrayendo con pautas lógicas las guías y bases para elaborar los algoritmos que los solucionan, a partir de la documentación y gráficos de análisis y diseño del problema.
- 1.2 Los problemas a resolver se descomponen, modularizándolos usando estrategias 'top-down' o 'bottom-up'.
- 1.3 Los problemas a resolver se modelizan usando la herramienta de la abstracción lógica, siguiendo algoritmos y técnicas tales como gestión y ordenación de archivos, búsqueda ordenada y secuencial, divide y vencerás, programación dinámica, algoritmos voraces, recursividad, búsquedas con retroceso, ramificación y poda, programación lineal y reducciones y algoritmos recursivos, entre otros, para abordar su resolución de forma simplificada.
- 1.4 Los algoritmos se representan, usando herramientas al efecto de modularización y prototipado tales como pseudocódigo y diagramas de flujo.

2. Elaborar programas para Inteligencia Artificial bajo el paradigma de Programación Orientada a Objetos (POO) en lenguajes de uso propio del sector para aplicar los algoritmos elaborados, bajo las normas de calidad y las pautas de programación de la organización, siguiendo especificaciones de análisis y diseño.

- 2.1 Las clases definidas, sus atributos, métodos e interfaces se codifican utilizando el lenguaje de programación elegido y las técnicas metodológicas orientadas a objetos, interpretando los modelos desarrollados en el análisis.
- 2.2 Los tipos y estructuras de datos básicos tales como aquellos que representan caracteres, números y valores lógicos, entre otros y los tipos complejos como cadenas, vectores, tablas multidimensionales, árboles, grafos, pilas, colas, tablas hash y colecciones se gestionan, aplicándolos a los atributos, parámetros de métodos y/o clases, usando sus métodos predefinidos, siguiendo los algoritmos para la resolución de los problemas recibidos e interpretando los modelos desarrollados en el análisis.
- 2.3 Las herramientas estructuradas de programación tales como asignaciones, condiciones, bucles e iteraciones, entre otras, se aplican en los métodos, previa definición de sus parámetros de entrada y/o salida, siguiendo los algoritmos elaborados para resolver los problemas planteados.
- 2.4 Las relaciones entre clases se codifican, a partir de la documentación de análisis y diseño, derivando o extendiendo clases hija de las clases base o padre asociadas, manteniendo las restricciones, visibilidad,

multiplicidades, herencia y otras características especificadas, tales como gestión del envío de mensajes e implementándolas con las técnicas y utilidades de la programación orientada a objetos tales como sobrecarga de constructores y métodos, uso de tipos abstractos y polimorfismo, entre otros.

- 2.5 Los objetos se determinan, interpretando las especificaciones establecidas en el análisis y diseño y los algoritmos, programando los constructores e instanciando las clases definidas.
- 2.6 El control de errores se implementa en el código, usando las sentencias y mecanismo propios del lenguaje tales como 'Try-Catch' u otros, de acuerdo a las normas de calidad de la organización.
- 2.7 Las herramientas de desarrollo y depuración se emplean para facilitar el proceso de generación del código, detectando y corrigiendo errores en el código.
- 2.8 Los componentes y librerías de terceros se utilizan, seleccionándolos según las necesidades, interpretando su documentación e implementando su uso, para reutilizar código y reducir el tiempo de desarrollo.
- 2.9 El acceso y manipulación de datos estructurados y no estructurados externos se programa, utilizando librerías disponibles e integrándolas en la aplicación, codificando programas a partir de ellas.
- 2.10 El plan de pruebas se ejecuta, siguiendo los protocolos y los criterios de calidad definidos en la organización, discriminando los tipos de prueba, resolviendo los problemas encontrados en la implementación, consultando la documentación técnica.
- 2.11 Los elementos elaborados susceptibles de ser compartidos, se integran en paquetes y librerías para su uso posterior en futuras aplicaciones.

3. Programar sistemas de redes neuronales de extracción del conocimiento de la Inteligencia Artificial para la predicción de tendencias o la organización y descripción de datos, codificando programas mediante técnicas y algoritmos al efecto bajo supervisión del científico de datos responsable.

- 3.1 Los algoritmos de aprendizaje automático supervisados (predictivos) se programan, usando redes neuronales con el objeto de predecir un dato (variable objetivo) desconocido a priori, a partir de otros datos conocidos.
- 3.2 Los algoritmos de aprendizaje automático no supervisados se programan, usando redes neuronales, buscando descubrir patrones (estructuras) y tendencias en los datos, con el objetivo de organizar los datos de la manera especificada para el proyecto o describir su estructura.
- 3.3 El código se programa siguiendo métodos de cuantización ('clustering') de vectores tales como 'K-mean', con el enfoque de examinar y presentar, datos complejos de manera más simple, para que el científico de datos responsable los analice.

- 3.4 El espacio de entrada de las muestras de entrenamiento se representa, programando código que use técnicas tales como 'mapas auto organizativos de Kohonen', 'SOM-Self-Organizing Maps' (SOM), para discretizarlo.
- 3.5 Las redes asociativas se simulan, programando código en base a técnicas tales como 'Programación de Redes de Hopfield' o redes recursivas recurrentes para garantizar la convergencia a un mínimo local.
- 3.6 La resolución de problemas que no son linealmente separables se explora programando técnicas tales como Redes Neuronales con Perceptrones Multicapa u otras.

4. Programar software para la extracción de patrones preparatorios aplicables en aprendizaje automático ('Machine Learning' -ML-) usando librerías predefinidas, para aplicar un análisis estadístico de los datos, bajo supervisión del científico de datos responsable.

- 4.1 El software de análisis de datos se codifica, a partir de librerías tales como Numpy en Jupyter, usando array multidimensional homogéneo.
- 4.2 Los datos se manipulan, previo análisis, usando librerías tales como Pandas que proporciona estructuras de datos y funcionalidades al efecto tales como series y DataFrames.
- 4.3 Los datos se representan, usando librerías tipo 'Plotter' tales como Matplotlib y Seaborn y programación de rectas y curvas de regresión de los datos de ajuste.
- 4.4 Los datos se integran, unificando y uniformando instancias provenientes de distintas fuentes, eliminando atributos redundantes y limpiando los datos, tratando los valores faltantes y el ruido de datos.
- 4.5 Los repositorios de información se filtran, descubriendo y extrayendo información siguiendo un proceso tal como 'Knowledge Discovery in Data bases' (KDD) para obtener todo el conocimiento posible.
- 4.6 Los algoritmos tales como clasificación, regresión lineal, regresión logística, uso de árboles de decisión, uso de polinomios y variabilidad se aplican al análisis del lenguaje natural (NLP) y redes sociales, escribiendo el código para extraer y clasificar información.

5. Elaborar la documentación del código desarrollado para sistemas de Inteligencia Artificial según los estándares de la organización, para garantizar su futuro mantenimiento.

- 5.1 La documentación tal como manuales, guías para desarrolladores, administradores o usuarios, entre otras se redacta de acuerdo a las especificaciones del proyecto, incluyendo información de utilidad según el destinatario.
- 5.2 La documentación correspondiente a cada componente se redacta, de acuerdo con las normas y modelos incluidos en el diseño de la aplicación.

- 5.3 Los comentarios, identificadores con nombres autoexplicativos (mnemotécnicos), tabulaciones u otros, se añaden al código, documentándolo para la mejor comprensión del programa.
- 5.4 La documentación de un componente desarrollado se genera mediante las herramientas de producción automática de documentación a partir del texto incluido en el componente software.
- 5.5 Los procedimientos de ejecución de las pruebas unitarias y de regresión, así como los resultados de las mismas se documentan de acuerdo a los estándares definidos por la organización.

b) Especificaciones relacionadas con el “saber”.

La persona candidata, en su caso, deberá demostrar que posee los conocimientos técnicos (conceptos y procedimientos) que dan soporte a las actividades profesionales implicadas en las realizaciones profesionales de la **UC2497_3: Desarrollar componentes software específicos para sistemas de inteligencia artificial basados en aprendizaje automático**. Estos conocimientos se presentan agrupados a partir de las actividades profesionales que aparecen en cursiva y negrita:

1. Modelización de algoritmos de carácter general y específicos de Inteligencia Artificial

- Técnicas de abstracción de problemas reales para su traslado a la elaboración de algoritmos.
- Interpretación de gráficos y la documentación de análisis y diseño para la elaboración de algoritmos.
- Técnicas de modularización. 'Top-down' y 'bottom-up'.
- Herramientas de abstracción lógica para modelizar. Algoritmos y técnicas. Gestión y ordenación de archivos, búsqueda ordenada y secuencial, divide y vencerás, programación dinámica, algoritmos voraces, recursividad, búsquedas con retroceso, ramificación y poda, programación lineal y reducciones, algoritmos recursivos y otros.
- Técnicas de representación de algoritmos: pseudocódigo y diagramas de flujo.

2. Resolución de problemas generales y específicos de Inteligencia Artificial aplicando Programación Orientada a Objetos

- Clases. Tipología. Clase base, clase derivada, clase abstracta.
- Relaciones entre clases. Herencia. Visibilidad. Multiplicidad.
- Atributos y operaciones.
- Sobrecarga. Polimorfismo.
- Interfaces. Implementación y aplicación.
- Constructores.
- Tipos de datos básicos o elementales. Numéricos, alfanuméricos y lógicos ('boolean').

- Tipos y estructuras de datos complejos. Cadenas, vectores, tablas multidimensionales, árboles, grafos, conjuntos, pilas, colas, tablas hash y colecciones.
- Paso de mensajes.
- Sentencias de la programación estructurada: asignaciones, condiciones, bucles e iteraciones.

3. Mecanismos de control, gestión en integración en Programación Orientada a Objetos aplicada a la Inteligencia Artificial

- Control de errores.
- Herramientas de desarrollo y depuración.
- Componentes de terceros. Complementos y librerías.
- Librerías y mecanismos de acceso y manipulación de datos estructurados y no estructurados externos.
- Procedimientos a seguir para elaborar un plan de pruebas. Tipos de prueba.
- Empaquetado de código en librerías. Refactorización.

4. Interpretación de diagramas de modelización de problemas

- Estándares. 'Unified Modelling Language' (UML).
- Interpretación de diagramas que representan el sistema de manera estática. Diagramas de Estructura Estática. Diagramas de Clases.
- Interpretación de diagramas de interacción. Interpretación de diagramas de Casos de Uso.
- Interpretación de diagramas que representan la interacción organizada. Diagramas de Casos de Colaboración.
- Interpretación de diagramas de Modelado Dinámico. Diagramas de Actividades. Diagramas de Secuencia, Diagramas de Colaboración, Diagramas de Estados, Diagramas de Casos de Uso, Diagramas de Actividades.

5. Predicción de tendencias/organización y descripción de datos mediante redes neuronales usando Programación Orientada a Objetos

- Modelos de aprendizaje de las redes neuronales (RRN). Supervisados y no supervisados.
- Métodos de cuantización ('clustering') de vectores. 'K-means'.
- Técnicas de representación del espacio de entrada de las muestras de entrenamiento. Mapas auto organizativos de Kohonen SOM ('SOM-Self-Organizing Maps).
- Técnicas de simulación de redes asociativas. 'Programación de Redes de Hopfield', redes recursivas recurrentes.
- Técnicas de resolución de problemas no linealmente separables. Programación de Redes Neuronales con Perceptrones Multicapa.

6. Técnicas estadísticas de análisis de datos usando Programación Orientada a Objetos en Inteligencia Artificial

- Librerías de análisis de datos tales como 'Numpy' en 'Jupyter' u otras.
- Librerías de manipulación y análisis tales como 'Pandas' u otras.
- Librerías de representación tipo 'Plotter' tales como 'Matplotlib', 'Seaborn' u otras.

- Programación de rectas y curvas de regresión de los datos de ajuste.
- Mecanismos de integración, unificación y uniformización de datos e instancias provenientes de distintas fuentes.
- Técnicas de filtrado de repositorios de información para descubrir y extraer información. 'Knowledge Discovery in Data bases' (KDD), Análisis del lenguaje natural (NLP) y de las redes sociales. Clasificación, regresión lineal, regresión logística, uso de árboles de decisión, uso de polinomios y variabilidad.

c) Especificaciones relacionadas con el “saber estar”.

La persona candidata debe demostrar la posesión de actitudes de comportamiento en el trabajo y formas de actuar e interactuar, según las siguientes especificaciones:

- Responsabilizarse del trabajo que desarrolla y del cumplimiento de los objetivos.
- Demostrar cierto grado de autonomía en la resolución de contingencias relacionadas con su actividad.
- Comunicarse eficazmente con las personas adecuadas en cada momento, respetando los canales establecidos en la organización.
- Adaptarse a la organización, a sus cambios organizativos y tecnológicos, así como a situaciones o contextos nuevos.
- Adoptar actitudes posturales adecuadas en el entorno de trabajo.
- Mostrar una actitud de respeto hacia los compañeros, procedimientos y normas de la empresa.
- Cumplir las medidas que favorezcan el principio de igualdad de trato y de oportunidades entre hombres y mujeres.
- Valorar el talento y el rendimiento profesional con independencia del sexo.
- Aplicar de forma efectiva el principio de igualdad de trato y no discriminación en las condiciones de trabajo entre mujeres y hombres.

1.2. Situaciones profesionales de evaluación y criterios de evaluación.

La situación profesional de evaluación define el contexto profesional en el que se tiene que desarrollar la misma. Esta situación permite al evaluador o evaluadora obtener evidencias de competencia de la persona candidata que incluyen, básicamente, todo el contexto profesional de la Unidad de Competencia implicada.

Así mismo, la situación profesional de evaluación se sustenta en actividades profesionales que permiten inferir competencia profesional respecto a la práctica totalidad de realizaciones profesionales de la Unidad de Competencia.

Por último, indicar que la situación profesional de evaluación define un contexto abierto y flexible, que puede ser completado por las CC.AA.,

cuando éstas decidan aplicar una prueba profesional a las personas candidatas.

En el caso de la “UC2497_3: Desarrollar componentes software específicos para sistemas de inteligencia artificial basados en aprendizaje automático”, se tiene una situación profesional de evaluación y se concreta en los siguientes términos:

1.2.1. Situación profesional de evaluación.

a) Descripción de la situación profesional de evaluación.

En esta situación profesional, la persona candidata demostrará la competencia requerida para desarrollar componentes software específicos para sistemas de inteligencia artificial basados en aprendizaje automático, según orden de trabajo y especificaciones técnicas. Esta situación comprenderá al menos las siguientes actividades:

1. Modelizar algoritmos de carácter general y específicos de Inteligencia Artificial.
2. Elaborar programas para la Inteligencia Artificial bajo el paradigma de Programación Orientada a Objetos (POO).
3. Programar sistemas de redes neuronales de extracción de conocimiento.
4. Programar software para la extracción de patrones preparatorios aplicables en el aprendizaje automático.

Condiciones adicionales:

- Se dispondrá de equipamientos, productos específicos y ayudas técnicas requeridas por la situación profesional de evaluación.
- Se comprobará la capacidad del candidato o candidata en respuesta a contingencias.
- Se asignará un tiempo total para que el candidato o la candidata demuestre su competencia en condiciones de estrés profesional.

b) Criterios de evaluación asociados a la situación de evaluación.

Cada criterio de evaluación está formado por un criterio de mérito significativo, así como por los indicadores y escalas de desempeño competente asociados a cada uno de dichos criterios.

En la situación profesional de evaluación, los criterios de evaluación se especifican en el cuadro siguiente:

<i>Criterios de mérito</i>	<i>Indicadores de desempeño competente</i>
<i>Eficacia en la modelización de algoritmos de carácter general y específicos de Inteligencia Artificial</i>	<ul style="list-style-type: none">- Interpretación de los problemas a resolver, abstrayendo con pautas lógicas las guías y bases para elaborar los algoritmos que los solucionan, a partir de la documentación y gráficos de análisis y diseño del problema.- Descomposición de los problemas a resolver, modularizándolos usando estrategias 'top-down' o 'bottom-up'.- Modelización de los problemas a resolver usando la herramienta de la abstracción lógica, siguiendo algoritmos y técnicas tales como gestión y ordenación de archivos, búsqueda ordenada y secuencial, divide y vencerás, programación dinámica, algoritmos voraces, recursividad, búsquedas con retroceso, ramificación y poda, programación lineal y reducciones y algoritmos recursivos, entre otros, para abordar su resolución de forma simplificada.- Representación de los algoritmos, usando herramientas al efecto de modularización y prototipado tales como pseudocódigo y diagramas de flujo. <p><i>El desempeño competente requiere el cumplimiento total de este criterio de mérito.</i></p>
<i>Calidad en la elaboración de programas para la Inteligencia Artificial bajo el paradigma de Programación Orientada a Objetos (POO).</i>	<ul style="list-style-type: none">- Codificación de las clases definidas, sus atributos, métodos e interfaces.- Gestión de los tipos y estructuras de datos básicos.- Aplicación de las herramientas estructuradas de programación.- Determinación de los objetos. <p><i>El umbral de desempeño competente está explicitado en la Escala A.</i></p>

<i>Idoneidad en la programación de sistemas de redes neuronales de extracción de conocimiento.</i>	<ul style="list-style-type: none">- Programación de los algoritmos de aprendizaje automático supervisados (predictivos).- Programación de los algoritmos de aprendizaje automático no supervisados.- Programación del código siguiendo métodos de cuantización ('clustering') de vectores.- Simulación de las redes asociativas. <p><i>El umbral de desempeño competente está explicitado en la Escala B.</i></p>
<i>Idoneidad en la programación de software para la extracción de patrones preparatorios aplicables en el aprendizaje automático.</i>	<ul style="list-style-type: none">- Codificación del software de análisis de datos.- Manipulación de datos, previo análisis.- Representación de los datos.- Integración de los los datos.- Aplicación de algoritmos. <p><i>El umbral de desempeño competente está explicitado en la Escala C.</i></p>
<i>Cumplimiento del tiempo asignado, considerando el que emplearía un o una profesional competente.</i>	<i>El desempeño competente permite sobrepasar el tiempo asignado hasta en un 25%</i>
<i>El desempeño competente requiere el cumplimiento, en todos los criterios de mérito, de la normativa aplicable en materia de prevención de riesgos laborales, protección medioambiental</i>	

Escala A

4	<p><i>Para la elaboración de programas para la Inteligencia Artificial bajo el paradigma de Programación Orientada a Objetos (POO), codifica las clases definidas, sus atributos, métodos e interfaces utilizando el lenguaje de programación elegido y las técnicas metodológicas orientadas a objetos, además de las relaciones entre clases. Gestiona los tipos y estructuras de datos básicos tales como aquellos que representan caracteres, números y valores lógicos, entre otros y los tipos complejos como cadenas, vectores, tablas multidimensionales, árboles, grafos, pilas, colas, tablas hash y colecciones, aplicándolos a los atributos, parámetros de métodos y/o clases, usando sus métodos predefinidos, siguiendo los algoritmos para la resolución de los problemas recibidos e interpretando los modelos desarrollados en el análisis. Aplica las herramientas estructuradas de programación tales como asignaciones, condiciones, bucles e iteraciones, entre otras, en los métodos, previa definición de sus parámetros de entrada y/o salida, siguiendo los algoritmos elaborados para resolver los problemas planteados. Determina los objetos, interpretando las especificaciones establecidas en el</i></p>
---	--

3	<p><i>análisis y diseño y los algoritmos, programando los constructores e instanciando las clases definidas y corrige posibles errores.</i></p> <p><i>Para la elaboración de programas para la Inteligencia Artificial bajo el paradigma de Programación Orientada a Objetos (POO), codifica las clases definidas, sus atributos, métodos e interfaces utilizando el lenguaje de programación elegido y las técnicas metodológicas orientadas a objetos, además de las relaciones entre clases. Gestiona los tipos y estructuras de datos básicos tales como aquellos que representan caracteres, números y valores lógicos, entre otros y los tipos complejos como cadenas, vectores, tablas multidimensionales, árboles, grafos, pilas, colas, tablas hash y colecciones, aplicándolos a los atributos, parámetros de métodos y/o clases, usando sus métodos predefinidos, siguiendo los algoritmos para la resolución de los problemas recibidos e interpretando los modelos desarrollados en el análisis. Aplica las herramientas estructuradas de programación tales como asignaciones, condiciones, bucles e iteraciones, entre otras, en los métodos, previa definición de sus parámetros de entrada y/o salida, siguiendo los algoritmos elaborados para resolver los problemas planteados. Determina los objetos, interpretando las especificaciones establecidas en el análisis y diseño y los algoritmos, programando los constructores e instanciando las clases definidas, pero comete pequeños errores que no afectan al resultado final.</i></p>
2	<p><i>Para la elaboración de programas para la Inteligencia Artificial bajo el paradigma de Programación Orientada a Objetos (POO), codifica las clases definidas, sus atributos, métodos e interfaces utilizando el lenguaje de programación elegido y las técnicas metodológicas orientadas a objetos, además de las relaciones entre clases. Gestiona los tipos y estructuras de datos básicos tales como aquellos que representan caracteres, números y valores lógicos, entre otros y los tipos complejos como cadenas, vectores, tablas multidimensionales, árboles, grafos, pilas, colas, tablas hash y colecciones, aplicándolos a los atributos, parámetros de métodos y/o clases, usando sus métodos predefinidos, siguiendo los algoritmos para la resolución de los problemas recibidos e interpretando los modelos desarrollados en el análisis. Aplica las herramientas estructuradas de programación tales como asignaciones, condiciones, bucles e iteraciones, entre otras, en los métodos, previa definición de sus parámetros de entrada y/o salida, siguiendo los algoritmos elaborados para resolver los problemas planteados. Determina los objetos, interpretando las especificaciones establecidas en el análisis y diseño y los algoritmos, programando los constructores e instanciando las clases definidas, pero comete grandes errores que afectan al resultado final.</i></p>
1	<p><i>No elabora programas para la Inteligencia Artificial bajo el paradigma de Programación Orientada a Objetos (POO).</i></p>

Nota: el umbral de desempeño competente corresponde a la descripción establecida en el número 3 de la escala.

Escala B

4	<p><i>Para la programación de sistemas de redes neuronales de extracción de conocimiento, programa los algoritmos de aprendizaje automático supervisados (predictivos), usando redes neuronales con el objeto de predecir un dato (variable objetivo) desconocido a priori, a partir de otros datos conocidos. Programa los algoritmos de aprendizaje automático no supervisados, usando redes neuronales, buscando descubrir patrones (estructuras) y tendencias en los datos, con el objetivo de organizar los</i></p>
---	--

	<p><i>datos de la manera especificada para el proyecto o describir su estructura. Programa el código siguiendo métodos de cuantización ('clustering') de vectores tales como 'K-mean', con el enfoque de examinar y presentar, datos complejos de manera más simple, para que el científico de datos responsable los analice. Simula las redes asociativas, programando código en base a técnicas tales como 'Programación de Redes de Hopfield' o redes recursivas recurrentes para garantizar la convergencia a un mínimo local y corrige posibles errores.</i></p>
3	<p><i>Para la programación de sistemas de redes neuronales de extracción de conocimiento, programa los algoritmos de aprendizaje automático supervisados (predictivos), usando redes neuronales con el objeto de predecir un dato (variable objetivo) desconocido a priori, a partir de otros datos conocidos. Programa los algoritmos de aprendizaje automático no supervisados, usando redes neuronales, buscando descubrir patrones (estructuras) y tendencias en los datos, con el objetivo de organizar los datos de la manera especificada para el proyecto o describir su estructura. Programa el código siguiendo métodos de cuantización ('clustering') de vectores tales como 'K-mean', con el enfoque de examinar y presentar, datos complejos de manera más simple, para que el científico de datos responsable los analice. Simula las redes asociativas, programando código en base a técnicas tales como 'Programación de Redes de Hopfield' o redes recursivas recurrentes para garantizar la convergencia a un mínimo local, pero comete grandes errores que no afectan al resultado final.</i></p>
2	<p><i>Para la programación de sistemas de redes neuronales de extracción de conocimiento, programa los algoritmos de aprendizaje automático supervisados (predictivos), usando redes neuronales con el objeto de predecir un dato (variable objetivo) desconocido a priori, a partir de otros datos conocidos. Programa los algoritmos de aprendizaje automático no supervisados, usando redes neuronales, buscando descubrir patrones (estructuras) y tendencias en los datos, con el objetivo de organizar los datos de la manera especificada para el proyecto o describir su estructura. Programa el código siguiendo métodos de cuantización ('clustering') de vectores tales como 'K-mean', con el enfoque de examinar y presentar, datos complejos de manera más simple, para que el científico de datos responsable los analice. Simula las redes asociativas, programando código en base a técnicas tales como 'Programación de Redes de Hopfield' o redes recursivas recurrentes para garantizar la convergencia a un mínimo local, pero cometen grandes errores que afectan al resultado final.</i></p>
1	<p><i>No programa sistemas de redes neuronales de extracción de conocimiento.</i></p>

Nota: el umbral de desempeño competente corresponde a la descripción establecida en el número 3 de la escala.

Escala C

4	<p><i>Para la programación de software para la extracción de patrones preparatorios aplicables en el aprendizaje automático, codifica el software de análisis de datos, a partir de librerías tales como Numpy en Jupyter, usando array multidimensional homogéneo. Manipula datos, previo análisis, usando librerías tales como Pandas que proporciona estructuras de datos y funcionalidades al efecto tales como series y DataFrames. Representa los datos, usando librerías tipo 'Plotter' tales como Matplotlib y Seaborn y programación de rectas y curvas de regresión de los datos de ajuste. Integra los los datos, unificando y uniformando instancias provenientes de distintas fuentes, eliminando atributos redundantes y limpiando los datos, tratando los valores faltantes y el ruido de datos. Aplica</i></p>
---	--

3	<p><i>los algoritmos tales como clasificación, regresión lineal, regresión logística, uso de árboles de decisión, uso de polinomios y variabilidad al análisis del lenguaje natural (NLP) y redes sociales, escribiendo el código para extraer y clasificar información y corrige posibles errores.</i></p> <p><i>Para la programación de software para la extracción de patrones preparatorios aplicables en el aprendizaje automático, codifica el software de análisis de datos, a partir de librerías tales como Numpy en Jupyter, usando array multidimensional homogéneo. Manipula datos, previo análisis, usando librerías tales como Pandas que proporciona estructuras de datos y funcionalidades al efecto tales como series y DataFrames. Representa los datos, usando librerías tipo 'Plotter' tales como Matplotlib y Seaborn y programación de rectas y curvas de regresión de los datos de ajuste. Integra los los datos, unificando y uniformando instancias provenientes de distintas fuentes, eliminando atributos redundantes y limpiando los datos, tratando los valores faltantes y el ruido de datos. Aplica los algoritmos tales como clasificación, regresión lineal, regresión logística, uso de árboles de decisión, uso de polinomios y variabilidad al análisis del lenguaje natural (NLP) y redes sociales, escribiendo el código para extraer y clasificar información, pero comete pequeños errores que no afectan al resultado final.</i></p>
2	<p><i>Para la programación de software para la extracción de patrones preparatorios aplicables en el aprendizaje automático, codifica el software de análisis de datos, a partir de librerías tales como Numpy en Jupyter, usando array multidimensional homogéneo. Manipula datos, previo análisis, usando librerías tales como Pandas que proporciona estructuras de datos y funcionalidades al efecto tales como series y DataFrames. Representa los datos, usando librerías tipo 'Plotter' tales como Matplotlib y Seaborn y programación de rectas y curvas de regresión de los datos de ajuste. Integra los los datos, unificando y uniformando instancias provenientes de distintas fuentes, eliminando atributos redundantes y limpiando los datos, tratando los valores faltantes y el ruido de datos. Aplica los algoritmos tales como clasificación, regresión lineal, regresión logística, uso de árboles de decisión, uso de polinomios y variabilidad al análisis del lenguaje natural (NLP) y redes sociales, escribiendo el código para extraer y clasificar información, pero comete grande errores que afectan al resultado final.</i></p>
1	<p><i>No programa el software para la extracción de patrones preparatorios aplicables en el aprendizaje automático.</i></p>

Nota: el umbral de desempeño competente corresponde a la descripción establecida en el número 3 de la escala.

2. MÉTODOS DE EVALUACIÓN DE LA UNIDAD DE COMPETENCIA Y ORIENTACIONES PARA LAS COMISIONES DE EVALUACIÓN Y EVALUADORES/AS.

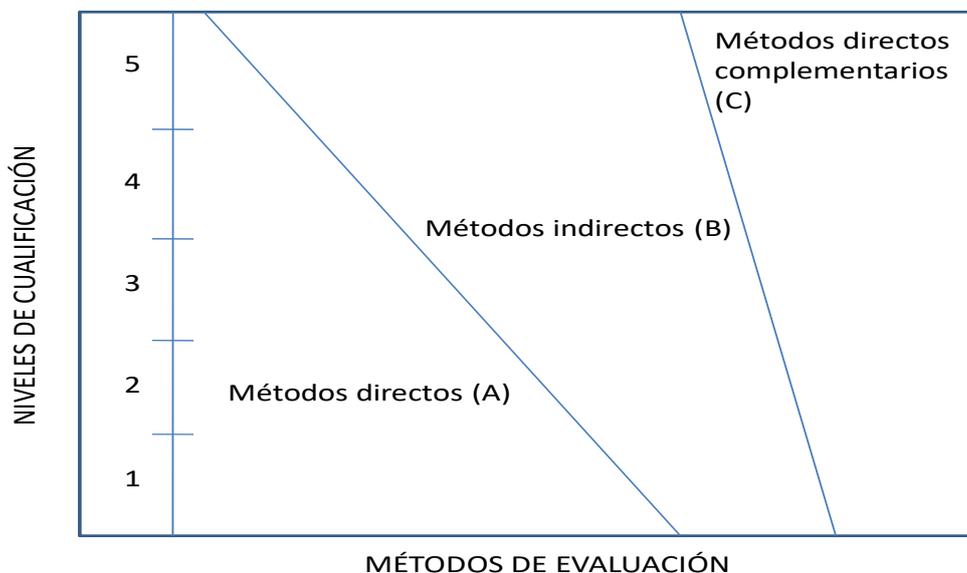
La selección de métodos de evaluación que deben realizar las Comisiones de Evaluación será específica para cada persona candidata, y dependerá fundamentalmente de tres factores: nivel de cualificación de la unidad de

competencia, características personales de la persona candidata y evidencias de competencia indirectas aportadas por la misma.

2.1. Métodos de evaluación y criterios generales de elección.

Los métodos que pueden ser empleados en la evaluación de la competencia profesional adquirida por las personas a través de la experiencia laboral, y vías no formales de formación son los que a continuación se relacionan:

- a) **Métodos indirectos:** Consisten en la valoración del historial profesional y formativo de la persona candidata; así como en la valoración de muestras sobre productos de su trabajo o de proyectos realizados. Proporcionan evidencias de competencia inferidas de actividades realizadas en el pasado.
- b) **Métodos directos:** Proporcionan evidencias de competencia en el mismo momento de realizar la evaluación. Los métodos directos susceptibles de ser utilizados son los siguientes:
 - Observación en el puesto de trabajo (A).
 - Observación de una situación de trabajo simulada (A).
 - Pruebas de competencia profesional basadas en las situaciones profesionales de evaluación (C).
 - Pruebas de habilidades (C).
 - Ejecución de un proyecto (C).
 - Entrevista profesional estructurada (C).
 - Preguntas orales (C).
 - Pruebas objetivas (C).



Fuente: Leonard Mertens (elaboración propia)

Como puede observarse en la figura anterior, en un proceso de evaluación que debe ser integrado (“holístico”), uno de los criterios de elección depende del nivel de cualificación de la UC. Como puede observarse, a menor nivel, deben priorizarse los métodos de observación en una situación de trabajo real o simulada, mientras que, a niveles superiores, debe priorizarse la utilización de métodos indirectos acompañados de entrevista profesional estructurada.

La consideración de las características personales de la persona candidata, debe basarse en el principio de equidad. Así, por este principio, debe priorizarse la selección de aquellos métodos de carácter complementario que faciliten la generación de evidencias válidas. En este orden de ideas, nunca debe aplicarse una prueba de conocimientos de carácter escrito a una persona candidata a la que se le aprecien dificultades de expresión escrita, ya sea por razones basadas en el desarrollo de las competencias básicas o factores de integración cultural, entre otras. Una conversación profesional que genere confianza sería el método adecuado.

Por último, indicar que las evidencias de competencia indirectas debidamente contrastadas y valoradas, pueden incidir decisivamente, en cada caso particular, en la elección de otros métodos de evaluación para obtener evidencias de competencia complementarias.

2.2. Orientaciones para las Comisiones de Evaluación y Evaluadores.

- a) Cuando la persona candidata justifique sólo formación formal y no tenga experiencia en el proceso de Planificar y determinar el proceso de decoración de vidrio mediante aplicaciones de color, se le someterá, al menos, a una prueba profesional de evaluación y a una entrevista profesional estructurada sobre la
- b) En la fase de evaluación siempre se deben contrastar las evidencias indirectas de competencia presentadas por la persona candidata. Deberá tomarse como referente la UC, el contexto que incluye la situación profesional de evaluación, y las especificaciones de los “saberes” incluidos en las dimensiones de la competencia. Se recomienda utilizar una entrevista profesional estructurada.
- c) Si se evalúa a la persona candidata a través de la observación en el puesto de trabajo, se recomienda tomar como referente los logros expresados en las realizaciones profesionales considerando el contexto expresado en la situación profesional de evaluación.
- d) Si se aplica una prueba práctica, se recomienda establecer un tiempo para su realización, considerando el que emplearía un o una profesional competente, para que el evaluado trabaje en condiciones de estrés profesional.
- e) Por la importancia del “saber estar” recogido en la letra c) del apartado 1.1 de esta Guía, en la fase de evaluación se debe comprobar la competencia de la persona candidata en esta dimensión particular, en los aspectos considerados.
- f) Esta Unidad de Competencia es de nivel "3" y sus competencias conjugan básicamente destrezas cognitivas y actitudinales. Por las características de estas competencias, la persona candidata ha de movilizar fundamentalmente sus destrezas cognitivas aplicándolas de forma competente a múltiples situaciones y contextos profesionales. Por esta razón, se recomienda que la comprobación de lo explicitado por la persona candidata se complemente con una prueba de desarrollo práctico, que tome como referente las actividades de la situación profesional de evaluación, todo ello con independencia del método de evaluación utilizado. Esta prueba se planteará sobre un contexto definido que permita evidenciar las citadas competencias, minimizando los recursos y el tiempo necesario para su realización, e implique el

cumplimiento de las normas de seguridad, prevención de riesgos laborales y medioambientales requeridas.

- g) Si se utiliza la entrevista profesional para comprobar lo explicitado por la persona candidata se tendrán en cuenta las siguientes recomendaciones:

Se estructurará la entrevista a partir del análisis previo de toda la documentación presentada por la persona candidata, así como de la información obtenida en la fase de asesoramiento y/o en otras fases de la evaluación.

La entrevista se concretará en una lista de cuestiones claras, que generen respuestas concretas, sobre aspectos que han de ser explorados a lo largo de la misma, teniendo en cuenta el referente de evaluación y el perfil de la persona candidata. Se debe evitar la improvisación.

El evaluador o evaluadora debe formular solamente una pregunta a la vez dando el tiempo suficiente de respuesta, poniendo la máxima atención y neutralidad en el contenido de las mismas, sin enjuiciarlas en ningún momento. Se deben evitar las interrupciones y dejar que la persona candidata se comunique con confianza, respetando su propio ritmo y solventando sus posibles dificultades de expresión.

Para el desarrollo de la entrevista se recomienda disponer de un lugar que respete la privacidad. Se recomienda que la entrevista sea grabada mediante un sistema de audio vídeo previa autorización de la persona implicada, cumpliéndose la ley de protección de datos.