



GUÍA DE EVIDENCIAS DEL ESTÁNDAR DE COMPETENCIAS PROFESIONALES

TRANSVERSAL

**“ECP0227_3: DESARROLLAR COMPONENTES SOFTWARE EN
LENGUAJES DE PROGRAMACIÓN ORIENTADOS A OBJETOS”**

1. ESPECIFICACIONES DE EVALUACIÓN DEL ESTÁNDAR DE COMPETENCIAS PROFESIONALES.

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 los elementos de la competencia (EC) e indicadores de calidad (IC) del ECP0227_3: DESARROLLAR COMPONENTES SOFTWARE EN LENGUAJES DE PROGRAMACIÓN ORIENTADOS A OBJETOS.

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 (Estándar de Competencias Profesionales (ECP) 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 en lenguajes de programación orientados a objetos, y que se indican a continuación:

Nota: A un dígito se indican las actividades profesionales expresadas en los elementos de la competencia del estándar de competencias profesionales, y dos dígitos las reflejadas en los indicadores de calidad.

1. Implementar los elementos software encomendados de modo que cumplan las especificaciones del diseño y los niveles de calidad establecidos por la organización para obtener los componentes orientados a objetos base de aplicaciones en diferentes tecnologías.

- 1.1 Los objetos, clases, atributos, métodos, relaciones e interfaces de los componentes del software se determinan, interpretando las especificaciones establecidas en el diseño.
- 1.2 Las inconsistencias del diseño detectadas se registran para su transmisión a la persona responsable por los cauces establecidos por la organización para garantizar la trazabilidad.
- 1.3 Las clases definidas, sus atributos, relaciones, métodos e interfaces se codifican utilizando el lenguaje de programación elegido, las técnicas metodológicas orientadas a objetos, las especificaciones recibidas, las normas de calidad y las pautas de programación de la organización.
- 1.4 El control de errores se implementa en el código de acuerdo a las normas de calidad de la organización.
- 1.5 Las herramientas de desarrollo y depuración se emplean para facilitar el proceso de generación del código, detectando y corrigiendo errores en el código.
- 1.6 Los problemas encontrados en la implementación se resuelven, consultando la documentación técnica, editada en castellano, en las lenguas cooficiales de las Comunidades Autónomas o en el idioma extranjero de uso más frecuente en el sector.
- 1.7 Los componentes de terceros se utilizan, seleccionándolos según las necesidades, interpretando su documentación e implementando su uso, para reutilizar código y reducir el tiempo de desarrollo.
- 1.8 Los componentes orientados a objeto se utilizan como base en el desarrollo de aplicaciones en diferentes tecnologías como programación web, dispositivos móviles, servicios en la nube, aprendizaje automático, u otras.

2. Manipular bases de datos a través de interfaces para integrar el lenguaje de programación con el lenguaje de acceso a datos en la construcción de una aplicación.

- 2.1 El conector de base de datos -con acceso directo, con mapeo objeto-relacional o el acceso directo a ficheros- se emplea en el programa, seleccionando una interfaz de programación de aplicaciones.
- 2.2 El inicio, configuración y cierre de una sesión en la base de datos se lleva a cabo con la interfaz de programación o bien utilizar una sesión abierta del grupo -pool- disponible.
- 2.3 Las operaciones de manipulación de datos utilizadas para el funcionamiento de la aplicación se ejecutan por medio de la interfaz de programación, garantizando la seguridad de la base de datos.

- 2.4 El interfaz de programación se utiliza para garantizar que los accesos a la base de datos finalizan las transacciones, asegurando su integridad y consistencia.

3. Probar los componentes software desarrollados para asegurar que cumplen las especificaciones definidas en el proyecto, garantizando su funcionamiento en producción.

- 3.1 El plan de pruebas se ejecuta, siguiendo los protocolos y los criterios de calidad definidos en la organización.
- 3.2 Las pruebas automatizadas se integran, ejecutándolas dentro de los procesos de integración continua y control de calidad.
- 3.3 Los datos de prueba de cada unidad funcional se concretan de forma que contemplen una variedad de posibilidades, incluyendo casos límite y conflictivos.
- 3.4 Las pruebas de cada unidad funcional se detallan de forma que cubran los caminos básicos que pueden surgir en su ejecución.
- 3.5 Los casos de prueba que involucran varios objetos se definen teniendo en cuenta los escenarios de interacción posibles para verificar que los objetos mantienen coherente su estado tras el intercambio de mensajes.
- 3.6 Las pruebas funcionales de alto nivel se efectúan, siguiendo los procesos definidos tanto de ejecución como documentación de forma que aseguren el cumplimiento de las especificaciones.
- 3.7 Las pruebas de puesta en producción se desarrollan para asegurar que los componentes soportan la carga de trabajo del entorno y su integración con servicios externos a los mismos -pasarelas de pago, servicios de internet u otros-.

4. Elaborar la documentación del código desarrollado según los estándares de la organización para garantizar su futuro mantenimiento.

- 4.1 La documentación -manuales, guías entre otras- para desarrolladores, administradores o usuarios se redacta de acuerdo a las especificaciones del proyecto.
- 4.2 La documentación correspondiente a cada componente se redacta de acuerdo con las normas y modelos incluidos en el diseño de la aplicación.
- 4.3 Los comentarios, identificadores con nombres autoexplicativos -mnemotécnicos-, tabulaciones u otros, se añaden al código, documentándolo para la mejor comprensión del programa-software-.
- 4.4 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.
- 4.5 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.

5. Gestionar los cambios en el código desarrollado para asegurar el cumplimiento de las normas de la organización y los niveles de calidad establecidos durante la fase de desarrollo y la de mantenimiento.

- 5.1 Los cambios en la implementación como consecuencia de cambios en las especificaciones se codifican, adecuando la estructura del código y manteniendo su calidad.
- 5.2 Las correcciones efectuadas como consecuencia de los errores o discrepancias encontrados en las pruebas se codifican en los componentes software afectados.
- 5.3 Las pruebas de regresión como consecuencia de cambios en las especificaciones se efectúan de acuerdo a los procedimientos establecidos para asegurar la funcionalidad previa.
- 5.4 La documentación del código se mantiene coherente con sus sucesivas modificaciones.

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 los elementos de la competencia del **ECP0227_3: Desarrollar componentes software en lenguajes de programación orientados a objetos**. Estos conocimientos se presentan agrupados a partir de las actividades profesionales que aparecen en cursiva y negrita:

1. Metodología de la programación orientada a objetos vinculada al desarrollo de componentes software en lenguajes de programación orientada a objetos

- Ciclo de desarrollo del software bajo el paradigma de orientación a objetos: Análisis, diseño y programación orientada a objetos.
- Proceso de construcción de software: Modularidad. Módulo en el paradigma orientado a objetos.
- Objetos como abstracciones de las entidades del mundo real que se quiere modelar.
- Descripción de objetos: Conjunto de datos que definen un objeto y conjunto de comportamientos que pueden solicitarse a los objetos.
- Comportamiento de un objeto: Concepto de mensaje.

2. Desarrollo de clases y objetos

- Clase: atributos, métodos y mecanismo de encapsulación. Interfaz de la clase. Clases y tipos de datos.
- Clasificación jerárquica de las clases. Clase raíz. Clases abstractas. Métodos virtuales. Redefinición de métodos.

- Polimorfismo. Tipos: Polimorfismo en tiempo de compilación (sobrecarga) y polimorfismo en tiempo de ejecución (ligadura dinámica). Objetos polimórficos. Comprobación estática y dinámica de tipos.
- Herencia. Herencia simple y múltiple. Reglas y características que definen una relación de herencia. Transmisión de atributos y métodos. Acceso a los atributos de una clase y acoplamiento entre las clases. Utilización de objetos. Creación de objetos en la herencia.
- Objetos: Estado, comportamiento e identidad. Mensajes. Tipos de métodos. Clasificación: Métodos de acceso, de selección o consulta, de construcción y de destrucción.
- Objetos como instancias de clase. Instancia actual.
- Concepto de programa en el paradigma orientado a objetos. POO = Objetos + Mensajes.
- Lenguajes de programación orientado a objetos, características. Lenguajes de programación basados en objetos y lenguajes de programación que utilizan objetos

3. Programación estructurada vinculada al desarrollo de componentes software en lenguajes de programación orientada a objetos

- Elementos: constantes, variables, operadores y expresiones.
- Tipos de datos: Datos simples, numéricos (enteros y reales), lógicos, carácter, cadena de caracteres, puntero o referencia a memoria; Datos estructurados, arrays, listas enlazadas, pilas y colas. Estructuras. Ficheros. Otras estructuras complejas: tablas hash e introducción a los árboles y grafos.
- Estructuras de control. Secuencial, condicional y de repetición.
- Funciones y procedimientos. Interfaz. Paso de parámetros: por valor y por referencia. Parámetros actuales y formales. Funciones: valor de retorno. Procedimientos. Ámbito de las variables. Almacenamiento de las variables.
- Llamadas a funciones y procedimientos. Llamadas a funciones y procedimientos incluidos en las clases. Llamadas calificadas y no calificadas (instancia actual). Paso de parámetros. Los atributos de la clase.
- Librerías. Estructura. Creación y utilización. Librerías de clases.
- Reutilización del software

4. Acceso a bases de datos y otras estructuras vinculado al desarrollo de componentes software en lenguajes de programación orientada a objetos

- Objetos de la base de datos.
- Integridad, consistencia y seguridad de los datos.
- Conexiones para el acceso a datos. Establecimiento, configuración y cierre de la conexión. Objetos de acceso a datos.
- Herramientas de acceso a datos proporcionadas por el entorno de programación.
- Herramientas de mapeado objeto-relacional (ORM)
- Sentencias del lenguaje de programación orientado a objetos para operar sobre las bases de datos. Integración de los objetos de la base de datos en el lenguaje de programación orientado a objetos. Integración de los tipos de datos propios del lenguaje de acceso a base de datos en el lenguaje de programación de la aplicación.
- Procedimientos almacenados.

- Transacciones distribuidas.
- Seguridad.
- Inyección de SQL.

5. Técnicas de pruebas de software

- Criterios de calidad. Métricas y estándares de calidad.
- Objetivos de las pruebas. Tipos de pruebas.
- Planificación de las pruebas: escenarios (datos -consultas, inserciones, borrados-, del sistema, de plataforma), casos de prueba.
- Proceso de pruebas. Pruebas de integración, de interfaces, de rendimiento y de seguridad.
- Herramientas para automatización de pruebas e integración continua.
- Gestión de errores.
- Gestión de cambios, concepto y pruebas de regresión.
- Análisis de resultados.
- Documentación de pruebas.

6. Mecanismos de gestión de memoria y tratamiento de errores

- Gestión automática de memoria.
- Construcción y destrucción de objetos. Objetos inalcanzables. Recolección de "basura".
- Métodos constructores y destructores.
- Definición de excepción. Fuentes de excepciones. Tratamiento de excepciones. Prevención de fallos. Excepciones definidas y lanzadas por el programador.
- Excepciones tratadas como objetos.

7. Técnicas de documentación

- Documentación de una aplicación, características, tipos: documentación técnica; guía de uso de la aplicación.
- Herramientas de documentación: características.
- Herramientas para generación de ayudas.
- Normativa aplicable y planificación de la actividad preventiva relativa a ergonomía, disposiciones mínimas de seguridad y salud en lugares de trabajo, entre otros.

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.
- Proponerse objetivos retadores que supongan un nivel de rendimiento y eficacia superior al alcanzado previamente.

- Demostrar cierto grado de autonomía en la resolución de contingencias relacionadas con su actividad.
- Aprender nuevos conceptos o procedimientos y aprovechar eficazmente la formación utilizando los conocimientos adquiridos.
- Adaptarse a la organización, a sus cambios organizativos y tecnológicos, así como a situaciones o contextos nuevos.
- Adoptar códigos de conducta tendentes a transmitir el contenido del principio de igualdad.

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 del Estándar de Competencias Profesionales implicado.

Así mismo, la situación profesional de evaluación se sustenta en actividades profesionales que permiten inferir competencia profesional respecto a la práctica totalidad de elementos de la competencia del Estándar de Competencias Profesionales.

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 del "ECP0227_3: Desarrollar componentes software en lenguajes de programación orientados a objetos", 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 en lenguajes de programación orientados a objetos, según orden de trabajo y especificaciones técnicas. Esta situación comprenderá al menos las siguientes actividades:

- 1. Implementar los elementos software encomendados.**
- 2. Manipular bases de datos a través de interfaces.**

3. Probar los componentes software desarrollados.

4. Elaborar la documentación del código desarrollado gestionando posibles cambios.

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:

Criterios de mérito	Indicadores de desempeño competente
<i>Eficacia en la implementación de los elementos software encomendados.</i>	<ul style="list-style-type: none">- Determinación de los objetos, clases, atributos, métodos, relaciones e interfaces de los componentes del software.- Codificación de las clases definidas, sus atributos, relaciones, métodos e interfaces.- Implementación del control de errores en el código.- Empleo de las herramientas de desarrollo y depuración para facilitar el proceso de generación del código.- Resolución de los problemas encontrados en la implementación. <p><i>El umbral de desempeño competente está explicitado en la Escala A.</i></p>

<p><i>Calidad en la manipulación de bases de datos a través de interfaces.</i></p>	<ul style="list-style-type: none">- Empleo del conector de base de datos.- Realización del inicio, configuración y cierre de una sesión en la base de datos con la interfaz.- Ejecución de las operaciones de manipulación de datos utilizadas para el funcionamiento de la aplicación por medio de la interfaz de programación.- Utilización del interfaz de programación para garantizar que los accesos a la base de datos finalizan las transacciones. <p><i>El umbral de desempeño competente está explicitado en la Escala B.</i></p>
<p><i>Eficiencia en la prueba de los componentes software desarrollados.</i></p>	<ul style="list-style-type: none">- Ejecución del plan de pruebas.- Integración de las pruebas automatizadas.- Definición de los casos de prueba que involucran varios objetos.- Realización de las pruebas funcionales de alto nivel.- Desarrollo de las pruebas de puesta en producción. <p><i>El umbral de desempeño competente está explicitado en la Escala C.</i></p>
<p><i>Idoneidad en la elaboración de la documentación del código desarrollado gestionando posibles cambios.</i></p>	<ul style="list-style-type: none">- Adición de los comentarios.- Generación de la documentación de un componente desarrollado.- Ejecución de los procedimientos de ejecución de las pruebas unitarias y de regresión.- Codificación de los cambios en la implementación como consecuencia de cambios en las especificaciones.- Codificación de las correcciones.- Mantenimiento de la documentación del código coherente. <p><i>El umbral de desempeño competente está explicitado en la Escala D.</i></p>
<p><i>Cumplimiento del tiempo asignado, considerando el que emplearía un o una profesional competente.</i></p>	<p><i>El desempeño competente permite sobrepasar el tiempo asignado hasta en un 25%</i></p>
<p><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></p>	

Escala A

4	<p><i>Para la implementación de los elementos software encomendados determina los objetos, clases, atributos, métodos, relaciones e interfaces de los componentes del software, interpretando las especificaciones establecidas en el diseño. Codifica las clases definidas, sus atributos, relaciones, métodos e interfaces utilizando el lenguaje de programación elegido, las técnicas metodológicas orientadas a objetos, las especificaciones recibidas, las normas de calidad y las pautas de programación de la organización. Implementa el control de errores en el código de acuerdo a las normas de calidad de la organización. Emplea las herramientas de desarrollo y depuración para facilitar el proceso de generación del código, detectando y corrigiendo errores en el código. Resuelve los problemas encontrados en la implementación, consultando la documentación técnica, editada en castellano, en las lenguas cooficiales de las Comunidades Autónomas o en el idioma extranjero de uso más frecuente en el sector y corrige posibles errores.</i></p>
3	<p>Para la implementación de los elementos software encomendados determina los objetos, clases, atributos, métodos, relaciones e interfaces de los componentes del software, interpretando las especificaciones establecidas en el diseño. Codifica las clases definidas, sus atributos, relaciones, métodos e interfaces utilizando el lenguaje de programación elegido, las técnicas metodológicas orientadas a objetos, las especificaciones recibidas, las normas de calidad y las pautas de programación de la organización. Implementa el control de errores en el código de acuerdo a las normas de calidad de la organización. Emplea las herramientas de desarrollo y depuración para facilitar el proceso de generación del código, detectando y corrigiendo errores en el código. Resuelve los problemas encontrados en la implementación, consultando la documentación técnica, editada en castellano, en las lenguas cooficiales de las Comunidades Autónomas o en el idioma extranjero de uso más frecuente en el sector, pero comete pequeños errores que no afectan al resultado final.</p>
2	<p><i>Para la implementación de los elementos software encomendados determina los objetos, clases, atributos, métodos, relaciones e interfaces de los componentes del software, interpretando las especificaciones establecidas en el diseño. Codifica las clases definidas, sus atributos, relaciones, métodos e interfaces utilizando el lenguaje de programación elegido, las técnicas metodológicas orientadas a objetos, las especificaciones recibidas, las normas de calidad y las pautas de programación de la organización. Implementa el control de errores en el código de acuerdo a las normas de calidad de la organización. Emplea las herramientas de desarrollo y depuración para facilitar el proceso de generación del código, detectando y corrigiendo errores en el código. Resuelve los problemas encontrados en la implementación, consultando la documentación técnica, editada en castellano, en las lenguas cooficiales de las Comunidades Autónomas o en el idioma extranjero de uso más frecuente en el sector, pero comete grandes errores que afectan al resultado final.</i></p>
1	<p><i>No implementa los elementos software encomendados.</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 manipulación de bases de datos a través de interfaces emplea el conector de base de datos -con acceso directo, con mapeo objeto-relacional o el acceso directo a ficheros- en el programa, seleccionando una interfaz de programación de aplicaciones. Realiza el inicio, configuración y cierre de una sesión en la base de datos con la interfaz de programación o bien utilizar una sesión abierta del grupo -pool- disponible. Ejecuta las operaciones de manipulación de datos utilizadas para el funcionamiento de la aplicación por medio de la interfaz de programación, garantizando la seguridad de la base de datos. Utiliza el interfaz de programación para garantizar que los accesos a la base de datos finalizan las transacciones, asegurando su integridad y consistencia y corrige posibles errores.</i></p>
3	<p>Para la manipulación de bases de datos a través de interfaces emplea el conector de base de datos -con acceso directo, con mapeo objeto-relacional o el acceso directo a ficheros- en el programa, seleccionando una interfaz de programación de aplicaciones. Realiza el inicio, configuración y cierre de una sesión en la base de datos con la interfaz de programación o bien utilizar una sesión abierta del grupo -pool- disponible. Ejecuta las operaciones de manipulación de datos utilizadas para el funcionamiento de la aplicación por medio de la interfaz de programación, garantizando la seguridad de la base de datos. Utiliza el interfaz de programación para garantizar que los accesos a la base de datos finalizan las transacciones, asegurando su integridad y consistencia, pero comete pequeños errores que no afectan al resultado final.</p>
2	<p><i>Para la manipulación de bases de datos a través de interfaces emplea el conector de base de datos -con acceso directo, con mapeo objeto-relacional o el acceso directo a ficheros- en el programa, seleccionando una interfaz de programación de aplicaciones. Realiza el inicio, configuración y cierre de una sesión en la base de datos con la interfaz de programación o bien utilizar una sesión abierta del grupo -pool- disponible. Ejecuta las operaciones de manipulación de datos utilizadas para el funcionamiento de la aplicación por medio de la interfaz de programación, garantizando la seguridad de la base de datos. Utiliza el interfaz de programación para garantizar que los accesos a la base de datos finalizan las transacciones, asegurando su integridad y consistencia, pero comete grandes errores que afectan al resultado final.</i></p>
1	<p><i>No manipula las bases de datos a través de interfaces.</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 prueba de los componentes software desarrollados ejecuta el plan de pruebas, siguiendo los protocolos y los criterios de calidad definidos en la organización. Integra las pruebas automatizadas, ejecutándolas dentro de los procesos de integración continua y control de calidad. Define los casos de prueba que involucran varios objetos teniendo en cuenta los escenarios de interacción posibles para verificar que los objetos mantienen coherente su estado tras el intercambio de mensajes. Realiza las pruebas funcionales de alto nivel, siguiendo los procesos definidos tanto de ejecución como documentación de forma que aseguren el cumplimiento de las especificaciones. Desarrolla las pruebas de puesta en producción para asegurar que los componentes soportan la carga de trabajo</i></p>
---	--

	<p>del entorno y su integración con servicios externos a los mismos -pasarelas de pago, servicios de internet u otros- y corrige posibles errores.</p>
3	<p>Para la prueba de los componentes software desarrollados ejecuta el plan de pruebas, siguiendo los protocolos y los criterios de calidad definidos en la organización. Integra las pruebas automatizadas, ejecutándolas dentro de los procesos de integración continua y control de calidad. Define los casos de prueba que involucran varios objetos teniendo en cuenta los escenarios de interacción posibles para verificar que los objetos mantienen coherente su estado tras el intercambio de mensajes. Realiza las pruebas funcionales de alto nivel, siguiendo los procesos definidos tanto de ejecución como documentación de forma que aseguren el cumplimiento de las especificaciones. Desarrolla las pruebas de puesta en producción para asegurar que los componentes soportan la carga de trabajo del entorno y su integración con servicios externos a los mismos -pasarelas de pago, servicios de internet u otros-, pero comete pequeños errores que no afectan al resultado final.</p>
2	<p>Para la prueba de los componentes software desarrollados ejecuta el plan de pruebas, siguiendo los protocolos y los criterios de calidad definidos en la organización. Integra las pruebas automatizadas, ejecutándolas dentro de los procesos de integración continua y control de calidad. Define los casos de prueba que involucran varios objetos teniendo en cuenta los escenarios de interacción posibles para verificar que los objetos mantienen coherente su estado tras el intercambio de mensajes. Realiza las pruebas funcionales de alto nivel, siguiendo los procesos definidos tanto de ejecución como documentación de forma que aseguren el cumplimiento de las especificaciones. Desarrolla las pruebas de puesta en producción para asegurar que los componentes soportan la carga de trabajo del entorno y su integración con servicios externos a los mismos -pasarelas de pago, servicios de internet u otros-, pero comete grandes errores que afectan al resultado final.</p>
1	<p>No prueba los componentes software desarrollados.</p>

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

Escala D

4	<p>Para la elaboración de la documentación del código desarrollado gestionando posibles cambios, añade los comentarios, identificadores con nombres autoexplicativos -mnemotécnicos-, tabulaciones u otros, al código, documentándolo para la mejor comprensión del programa-software-. Genera la documentación de un componente desarrollado mediante las herramientas de producción automática de documentación a partir del texto incluido en el componente software. Ejecuta los procedimientos de ejecución de las pruebas unitarias y de regresión, así como los resultados de las mismas de acuerdo a los estándares definidos por la organización. Codifica los cambios en la implementación como consecuencia de cambios en las especificaciones, adecuando la estructura del código y manteniendo su calidad. Codifica las correcciones efectuadas como consecuencia de los errores o discrepancias encontrados en las pruebas en los componentes software afectados. Mantiene la documentación del código coherente con sus sucesivas modificaciones y corrige posibles errores.</p>
3	

	<p><i>Para la elaboración de la documentación del código desarrollado gestionando posibles cambios, añade los comentarios, identificadores con nombres autoexplicativos -mnemotécnicos-, tabulaciones u otros, al código, documentándolo para la mejor comprensión del programa-software-. Genera la documentación de un componente desarrollado mediante las herramientas de producción automática de documentación a partir del texto incluido en el componente software. Ejecuta los procedimientos de ejecución de las pruebas unitarias y de regresión, así como los resultados de las mismas de acuerdo a los estándares definidos por la organización. Codifica los cambios en la implementación como consecuencia de cambios en las especificaciones, adecuando la estructura del código y manteniendo su calidad. Codifica las correcciones efectuadas como consecuencia de los errores o discrepancias encontrados en las pruebas en los componentes software afectados. Mantiene la documentación del código coherente con sus sucesivas modificaciones, pero comete pequeños errores que no afectan al resultado final.</i></p>
2	<p><i>Para la elaboración de la documentación del código desarrollado gestionando posibles cambios, añade los comentarios, identificadores con nombres autoexplicativos -mnemotécnicos-, tabulaciones u otros, al código, documentándolo para la mejor comprensión del programa-software-. Genera la documentación de un componente desarrollado mediante las herramientas de producción automática de documentación a partir del texto incluido en el componente software. Ejecuta los procedimientos de ejecución de las pruebas unitarias y de regresión, así como los resultados de las mismas de acuerdo a los estándares definidos por la organización. Codifica los cambios en la implementación como consecuencia de cambios en las especificaciones, adecuando la estructura del código y manteniendo su calidad. Codifica las correcciones efectuadas como consecuencia de los errores o discrepancias encontrados en las pruebas en los componentes software afectados. Mantiene la documentación del código coherente con sus sucesivas modificaciones, pero comete grandes errores que afectan al resultado final.</i></p>
1	<p><i>No elabora la documentación del código desarrollado ni gestiona posibles cambios.</i></p>

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

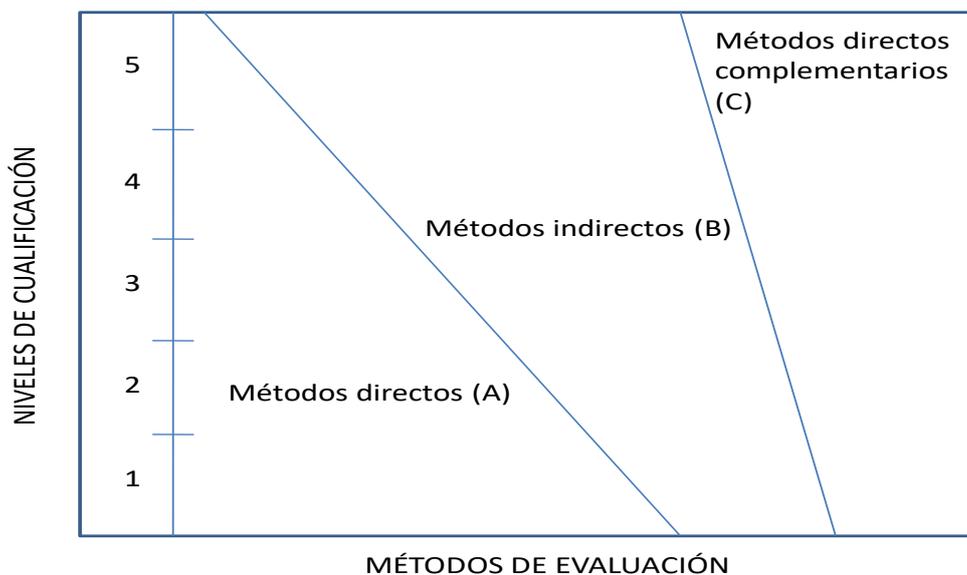
2. MÉTODOS DE EVALUACIÓN DEL ESTÁNDAR DE COMPETENCIAS PROFESIONALES Y ORIENTACIONES PARA LAS COMISIONES DE EVALUACIÓN Y EVALUADORES/AS.

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 del estándar de competencias profesionales, 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 del ECP. 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 Desarrollar componentes software en lenguajes de programación orientados a objetos, se le someterá, al menos, a una prueba profesional de evaluación y a una entrevista profesional estructurada sobre la dimensión relacionada con el "saber" y "saber estar" de la competencia profesional.
- b) En la fase de evaluación siempre se deben contrastar las evidencias indirectas de competencia presentadas por la persona candidata. Deberá tomarse como referente el ECP, 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 los elementos de la competencia 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) 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



UNIÓN EUROPEA
NextGenerationEU

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.