

## 2. RESULTADOS DEL PROCESO DE FORMACIÓN Y DE APRENDIZAJE

| CÓDIGO | COMPETENCIAS  |
|--------|---|
| C01    | Abordar problemas computacionales manejando apropiadamente la complejidad con el fin de resolver los problemas de manera efectiva y eficiente.  |
| C02    | Aplicar técnicas de desarrollo de software ágil y DevOps para resolver problemas en un contexto empresarial.  |
| C03    | Aplicar metodologías de desarrollo de software ágil en concordancia con la forma de trabajar demandada por las empresas actuales.   |
| C04    | Aplicar las prácticas más relevantes en el ámbito del desarrollo de software ágil, como el Desarrollo Dirigido por Tests (TDD), el Desarrollo Dirigido por Dominio (DDD), la programación por parejas o la automatización de pruebas.                         |
| C05    | Aplicar las prácticas más relevantes en el ámbito de la computación de la nube y la cultura DevOps, como el despliegue continuo, la contenedorización de aplicaciones, o el manejo y automatización de la infraestructura software por medio de código (IaC). |
| C06    | Aplicar principios de diseño software simple, como DRY, YAGNI, Ley de Demeter, etc.   |
| C07    | Diseñar e implementar arquitecturas centradas en el dominio, como la arquitectura hexagonal.  |
| C08    | Probar software de manera manual y también automática.  |
| C09    | Elaborar diferentes tipos de pruebas automáticas: unitarias, de integración, de sistema, etc.   |
| C10    | Automatizar despliegues de infraestructura por medio de herramientas como Terraform o Ansible.  |
| C11    | Trabajar con sistemas de control de versiones como Git o SVN.   |
| C12    | Gestionar dependencias de proyectos software por medio de herramientas como Maven o Gradle.   |
| CÓDIGO | HABILIDADES O DESTREZAS   |
| H01    | Utilizar buenas prácticas de ingeniería software para escribir código de calidad que sea fácilmente mantenible y comprensible.  |
| H02    | Desarrollar software de manera colaborativa aplicando los principios y valores de la cultura DevOps.  |
| H03    | Operar y desplegar aplicaciones en la nube (es decir, en entornos cloud como AWS o Azure).  |
| H04    | Diseñar, implementar y probar programas Python.   |
| H05    | Trabajar con código legado de manera efectiva.  |
| H06    | Manejar y configurar sistemas de monitorización como Elastic Stack.   |
| H07    | Manejar y configurar sistemas de alertas y visualización de datos como Graphana.  |

|               |   |
|---------------|---|
| H08           | Manejar y configurar bases de datos orientadas a métricas, como Prometheus.   |
| H09           | Manejar y configurar herramientas de contenedorización de aplicaciones, como Docker, Containerd o Podman.                                     |
| H10           | Manejar y configurar herramientas de orquestación de contenedores, como Kubernetes.   |
| H11           | Manejar y configurar entornos de integración y entrega continua, como Jenkins o GitLab.   |
| <b>CÓDIGO</b> | <b>CONOCIMIENTOS O CONTENIDOS</b>   |
| CC1           | Dominar las herramientas más relevantes en el ámbito de la computación de la nube y la cultura DevOps, como Docker, Terraform o Kubernetes.   |
| CC2           | Dominar de conceptos avanzados de programación, como la programación orientada a objetos, la programación funcional o los patrones de diseño. |
| CC3           | Seleccionar las estructuras de datos apropiadas que permitan producir programas eficientes.   |
| CC4           | Analizar la eficiencia de los programas tanto en términos de memoria como en tiempo de ejecución.   |
| CC5           | Mejorar la estructura del código aplicando técnicas de refactoring.   |