



**Guía de preparación**

Edición 201804

Copyright © EXIN Holding B.V. 2018. All rights reserved.  
EXIN® is a registered trademark.  
DevOps Master™ is a registered trademark.

No part of this publication may be published, reproduced, copied or stored in a data processing system or circulated in any form by print, photo print, microfilm or any other means without written permission by EXIN.



# Índice

1. Visión general	4
2. Requisitos del examen	8
3. Lista de Conceptos del examen	12
4. Bibliografía	15

# 1. Visión general

EXIN DevOps Master™ (DEVOPSM.SP)

## Alcance

EXIN DevOps Master™ es una certificación que acredita el conocimiento de un profesional sobre:

- Adopción de DevOps;
- Planificación, requerimientos y diseño;
- Desarrollo y Despliegue;
- Operación y Escalado;
- Fin de la Vida Útil.

## Resumen

El término DevOps es una contracción de las palabras inglesas "Development" (Desarrollo) y "Operations" (Operaciones). DevOps es un conjunto de prácticas recomendadas que enfatizan la colaboración y la comunicación entre los profesionales de TI (desarrolladores, administradores, operadores y personal de asistencia técnica) en el ciclo de vida de las aplicaciones y los servicios, lo que conduce a:

- Integración Continua: fusión de todas las copias de trabajo desarrolladas en una línea principal compartida varias veces al día
- Despliegue Continuo: publicación de versiones de forma continua o con la máxima frecuencia posible
- Retroalimentación Continua: búsqueda de retroalimentación de las partes interesadas durante todas las etapas del ciclo de vida

DevOps cambia la forma en la que las personas piensan sobre su trabajo; DevOps valora la diversidad del trabajo realizado, respalda los procesos intencionados que aumentan la velocidad a la que las empresas crean valor y mide el efecto del cambio técnico y social. DevOps es una manera de pensar y una manera de trabajar que permite a las personas y a las empresas desarrollar y mantener procedimientos de trabajo sostenibles.

Un DevOps satisfactorio consiste en:

- Promover una cultura libre de culpa en la que se compartan historias y se desarrolle la empatía para conseguir que las personas y los equipos desempeñen sus funciones de forma eficaz y duradera;
- Proporcionar aplicaciones y servicios al negocio según el modelo Just-in-Time (JiT);
- Garantizar la continuidad de los servicios de TI mediante una aproximación a las necesidades de negocio basada en el riesgo;
- Gestionar el ciclo de vida completo de las aplicaciones y los servicios, incluidas las condiciones de fin de la vida útil.

Esta certificación pretende aportar competencias prácticas al conocimiento con el fin de que una persona con la certificación EXIN DevOps Master™ pueda implementar el modelo DevOps de forma satisfactoria en un equipo y promover sus principios en la empresa.

Esta certificación ha sido desarrollada en colaboración con expertos en el sector DevOps.

## Contexto

El programa DevOps de EXIN:



## Grupo objetivo

DevOps es un modelo conocido en el campo del desarrollo de software, pero sus principios se pueden aplicar en los proyectos de creación o modificación de servicios de TI, así como en otros proyectos. La formación y la certificación de DevOps Master™ están dirigidas a todos los profesionales interesados en facilitar las prácticas DevOps en sus organizaciones.

La certificación EXIN DevOps Master™ está destinada a cualquier persona que trabaje en un equipo de DevOps o en una empresa que se esté planteando la transición a una metodología de trabajo DevOps. Entre las personas a las que está dirigida la certificación se incluyen: propietarios de productos, Agile Scrum Masters, gestores de proyectos, responsables de pruebas y aseguramiento de la calidad, directores de servicios de TI, propietarios de procesos y profesionales especializados en Lean IT.

## Requisitos para la certificación

1. Finalización con éxito de un curso de formación EXIN DevOps Master™ acreditado por EXIN (incluidas las tareas prácticas);
2. Realización satisfactoria del examen de DevOps Master™.

## Detalles del examen

Tipo de examen:	Questions à choix multiples
Número de preguntas:	50
Calificación mínima para el aprobado:	65%
Consulta de libro o apuntes:	No
Equipos electrónicos permitidos:	No
Duración del examen:	120 minutos

En este examen se aplican las normas de examen de EXIN.

## Nivel de Bloom

La certificación EXIN DevOps Master™ evalúa a los candidatos en los niveles 3 y 4 de Bloom de acuerdo con la taxonomía revisada de Bloom:

- Nivel 3 de Bloom: aplicación. Demuestra que los candidatos tienen la capacidad de utilizar la información en un contexto diferente al contexto donde la aprendieron. El objetivo de este tipo de preguntas es demostrar que el candidato es capaz de resolver problemas en situaciones nuevas aplicando el conocimiento adquirido, hechos, técnicas y normas de una forma diferente o nueva. En la pregunta se suele incluir una explicación del contexto.
- Nivel 4 de Bloom: análisis. Demuestra que los candidatos tienen la capacidad de fragmentar la información adquirida con el fin de entenderla. Este nivel de Bloom se evalúa principalmente en las tareas prácticas. El propósito de las tareas prácticas es demostrar que el candidato es capaz de examinar y fragmentar la información determinando los motivos o causas, sacar conclusiones y encontrar pruebas que respalden las generalizaciones.

## Formación

La formación es una parte obligatoria de la certificación. Se espera que los candidatos dispongan de los conocimientos básicos sobre los principios de DevOps y sobre los conceptos de las metodologías Lean y Agile. Estos conocimientos se pueden adquirir mediante:

- la certificación en EXIN DevOps Professional;
- leyendo "The Phoenix Project" (consulte la bibliografía).

## Horas de contacto

El número mínimo de horas de contacto para este curso de formación es de 24. Estas horas de contacto incluyen las prácticas, la preparación de exámenes y pausas breves. El número de horas no incluye el tiempo dedicado a los ejercicios, la logística relacionada con la sesión de examen, la sesión de examen o las pausas para almorzar.

Las directrices para las tareas prácticas están disponibles en PartnerNet. Se recomienda un máximo de 8 horas para completarlas. Las tareas prácticas pueden realizarse al margen de la formación. Los formadores podrán escoger entre las opciones siguientes:

- La simulación de GamingWorks (el formador necesita tener una acreditación de GamingWorks).
- El juego del libro: Introduction to DevOps with Chocolate, LEGO and Scrum Game.<sup>1</sup>
- Elaborar sus propias tareas prácticas.

La duración de la formación puede variar en función de la cantidad de trabajo de las tareas prácticas realizadas. Entre otras posibilidades se incluyen las siguientes:

- 2 jornadas de formación teórica en el aula y 1 jornada de formación práctica en el aula.
- 2 jornadas de formación teórica virtual y 8 horas de tareas prácticas en el lugar de trabajo; el candidato estudia mediante aprendizaje electrónico y prepara las tareas prácticas; el formador evalúa el trabajo realizado en las tareas prácticas.
- 2 jornadas de formación teórica en el aula y 8 horas de tareas prácticas en el lugar de trabajo; el formador evalúa el trabajo realizado en las tareas prácticas.

### Indicación de la carga de estudio

120 horas, en función de los conocimientos previos. La matriz bibliográfica del capítulo 4. *Literatura* en esta Guía de Preparación hace referencia al conjunto de conocimientos que se evaluará en el examen.

### Proveedor de la formación

Puede consultar una lista de nuestros proveedores de formación acreditados en [www.exin.com](http://www.exin.com).

## 2. Requisitos del examen

Los requisitos del examen se detallan en las especificaciones del mismo. La tabla que se muestra a continuación enumera los temas incluidos en el módulo (requisitos del examen) y los puntos que integran los mismos (especificaciones del examen).

Exigencias de l'examen	Spécifications de l'examen	Importance
<b>1. Adopción de DevOps</b>		<b>28%</b>
	1.1 La Mentalidad DevOps y los Beneficios	10%
	1.2 Cultura Organizativa	12%
	1.3 Principios y Conceptos	6%
<b>2. Planificación, requerimientos y diseño</b>		<b>18%</b>
	2.1 Gestión del Ciclo de vida de Aplicaciones o Servicios	4%
	2.2 Plan de Proyecto (Defining Scope) y Gestión Visual	4%
	2.3 Diseño de Infraestructuras y Arquitectura	4%
	2.4 Acuerdos y Requerimientos de Nivel de Servicio	2%
	2.5 Implementación de una Estrategia de Pruebas	4%
<b>3. Desarrollo y Despliegue</b>		<b>30%</b>
	3.1 Entrega e Integración Continuas	12%
	3.2 Pipeline de Despliegue	4%
	3.3 Despliegue Continuo	4%
	3.4 Ji-Kotei-Kanketsu, Ritmo, Work-in-Progress (WIP) y Flujo de una pieza	4%
	3.5 Automatización, Herramientas y Pruebas (Testing)	6%
<b>4. Operación y Escalado</b>		<b>22%</b>
	4.1 Gestión de Datos; Infraestructuras y Entornos; y Componentes y Dependencias	10%
	4.2 Gestión de Configuraciones y Control de Versiones	4%
	4.3 Cloud e Infraestructuras Estáticas	2%
	4.4 Continuidad de Negocio	2%
	4.5 Escalado	4%
<b>5. Fin de la Vida Útil</b>		<b>2%</b>
	5.1 Condiciones de Fin de la Vida Útil para un producto o servicio	2%
<b>Total</b>		<b>100%</b>



## Spécifications de l'examen

### 1. Adopción de DevOps

- 1.1 La Mentalidad DevOps y los Beneficios  
El candidato puede...
  - 1.1.1 analizar antipatrones DevOps en un escenario dado
  - 1.1.2 explicar los beneficios de DevOps
  - 1.1.3 explicar por qué DevOps encaja tan bien en los procesos de desarrollo de software actuales
  - 1.1.4 explicar por qué DevOps requiere de una forma especial de pensamiento para funcionar
  - 1.1.5 explicar cómo encaja DevOps con las prácticas Lean y Agile Scrum
- 1.2 Cultura Organizativa  
El candidato puede...
  - 1.2.1 explicar la importancia de los cuatro pilares para un DevOps efectivo (Colaboración, Afinidad, Herramientas y Escalado)
  - 1.2.2 analizar un escenario en busca de carencias en alguno de los elementos que componen la mentalidad DevOps
  - 1.2.3 explicar cómo crear un equipo a partir de un grupo de personas fomentando la colaboración, la mentalidad DevOps, la empatía y la confianza
  - 1.2.4 analizar una situación en la que haya ideas erróneas sobre la colaboración y sugerir métodos que resuelvan el problema
  - 1.2.5 analizar una situación en la que exista la necesidad de gestión de conflictos y plantear la mejor solución
  - 1.2.6 explicar cómo la gestión de recursos humanos puede fomentar la diversidad y el impacto beneficioso de ésta sobre la organización
- 1.3 Principios y Conceptos  
El candidato puede...
  - 1.3.1 explicar el uso y la utilidad de las diferentes metodologías de desarrollo de software (Waterfall, Agile, Scrum) y sus principios básicos
  - 1.3.2 explicar el uso y la utilidad de las diferentes metodologías utilizadas en operaciones (IT Service Management (Gestión de Servicios TI))
  - 1.3.3 explicar el uso y la utilidad de los métodos Lean

### 2. Planificación, requerimientos y diseño

- 2.1 Gestión del Ciclo de vida de Aplicaciones o Servicios  
El candidato puede...
  - 2.1.1 explicar cómo DevOps añade valor a la Gestión moderna del Ciclo de Vida de las Aplicaciones
  - 2.1.2 explicar cómo DevOps mejora la experiencia del cliente cuando se usa para la Gestión del Ciclo de Vida del Servicio
- 2.2 Plan de Proyecto (Defining Scope) y Gestión Visual  
El candidato puede...
  - 2.2.1 explicar cómo se determina el ámbito de un proyecto DevOps
  - 2.2.2 explicar por qué la Gestión Visual de un proyecto DevOps facilita las prácticas DevOps
- 2.3 Diseño de Infraestructuras y Arquitectura  
El candidato puede...
  - 2.3.1 explicar cómo DevOps cambia o influye en el diseño de infraestructuras y arquitecturas TI
  - 2.3.2 explicar por qué Computación en la nube y técnicas de virtualización facilitan la incorporación de Dev y Ops

- 2.4 Acuerdos y Requerimientos de Nivel de Servicio  
El candidato puede...
  - 2.4.1 explicar cómo DevOps cambia los Requisitos de Nivel de Servicio (SLR) y los Acuerdos de Nivel de Servicio (SLA)
- 2.5 Implementación de una Estrategia de Pruebas  
El candidato puede...
  - 2.5.1 explicar por qué y cómo debe ser modificada la Estrategia de Pruebas cuando se hace la transición a DevOps
  - 2.5.2 analizar y comprobar la integridad de las Historias de Usuario

### 3. Desarrollo y Despliegue

- 3.1 Entrega e Integración Continuas  
El candidato puede...
  - 3.1.1 explicar por qué la Entrega Continua es esencial para un DevOps efectivo
  - 3.1.2 analizar cómo integrar la Entrega Continua en un escenario
  - 3.1.3 analizar cómo resolver los problemas de Entrega Continua en un escenario
  - 3.1.4 explicar por qué la Integración Continua es esencial para un DevOps efectivo
  - 3.1.5 analizar cómo conseguir la Integración Continua en un escenario, con equipo distribuido o con un control de versiones distribuido
  - 3.1.6 analizar cómo resolver problemas en un escenario de Integración Continua
- 3.2 Pipeline de Despliegue  
El candidato puede...
  - 3.2.1 explicar la lógica de la anatomía de un pipeline de despliegue
  - 3.2.2 explicar cómo utilizar técnicas de *scripting* para la construcción (*build*) y el despliegue
- 3.3 Despliegue Continuo  
El candidato puede...
  - 3.3.1 explicar cómo se deben modificar el plan de versiones y de iteraciones para adaptarse a DevOps
  - 3.3.2 analizar cómo implementar el Despliegue Continuo en un escenario
- 3.4 Ji-Kotei-Kanketsu, Ritmo, Work-in-Progress (WIP) y Flujo de una pieza  
El candidato puede...
  - 3.4.1 explicar los conceptos *Ji-Kotei-Kanketsu*, Ritmo, *Work-in-Progress* (WIP) y Flujo de una pieza
  - 3.4.2 analizar un escenario buscando un problema al utilizar los conceptos de *Ji-Kotei-Kanketsu*, Ritmo, *Work-in-Progress* (WIP) y Flujo de una pieza y encontrar una solución adecuada
- 3.5 Automatización, Herramientas y Pruebas (Testing)  
El candidato puede...
  - 3.5.1 explicar por qué la automatización es importante para un DevOps efectivo
  - 3.5.2 explicar cómo utilizar las herramientas para facilitar DevOps en general
  - 3.5.3 explicar cómo utilizar las herramientas para apoyar la mentalidad y la cultura DevOps
  - 3.5.4 explicar por qué es fundamental que se automaticen las pruebas
  - 3.5.5 analizar un escenario y escoger la forma correcta de automatizar un test de aceptación

#### 4. Operación y Escalado

- 4.1 Gestión de Datos; Infraestructuras, y Entornos, y Componentes y Dependencias  
El candidato puede...
  - 4.1.1 explicar los problemas que se encuentran al manejar datos de bases de datos en entornos DevOps
  - 4.1.2 analizar un escenario donde se utiliza una base de datos en un entorno DevOps y dar la mejor solución al problema
  - 4.1.3 analizar un escenario e identificar la mejor forma de preparar la infraestructura para el despliegue, o bien para gestionarla después del despliegue
  - 4.1.4 analizar un escenario y sugerir estrategias comunes para la gestión de componentes
  - 4.1.5 explicar cómo gestionar dependencias
- 4.2 Gestión de Configuraciones y Control de Versiones  
El candidato puede...
  - 4.2.1 explicar por qué el control de versiones es clave para un DevOps efectivo
  - 4.2.2 explicar cómo mantener un control de versiones sobre datos, infraestructuras y componentes
  - 4.2.3 analizar un escenario y sugerir la mejor estrategia para atacar un problema de gestión de configuraciones
- 4.3 Cloud e Infraestructuras Estáticas  
El candidato puede...
  - 4.3.1 explicar cuándo es o no es necesario migrar a una infraestructura basada en la nube para un DevOps efectivo
  - 4.3.2 explicar cómo se debe gestionar la infraestructura basada en la nube en un entorno DevOps
- 4.4 Continuidad de Negocio  
El candidato puede...
  - 4.4.1 explicar cómo DevOps facilita las prácticas de Continuidad del Negocio
- 4.5 Escalado  
El candidato puede...
  - 4.5.1 analizar un escenario y explicar si es necesario realizar un escalado incremental o una reducción (*scale-up or scale-down*) en esa situación y por qué e identificar la mejor forma de hacerlo.
  - 4.5.2 analizar un escenario buscando qué ha ido mal en una situación de escalado e identificar una manera adecuada de resolver el problema
  - 4.5.3 explicar cómo las políticas sociales y de contratación apoyan el escalado de DevOps

#### 5. Fin de la Vida Útil

- 5.1 Condiciones de Fin de la Vida Útil para un producto o servicio  
El candidato puede...
  - 5.1.1 explicar qué condiciones se deben satisfacer antes de terminar un producto o servicio.

### 3. Lista de Conceptos del examen

En este capítulo figuran los términos y las abreviaturas que debería conocer el candidato.

*Tenga en cuenta que el conocimiento de estos términos por sí solo no bastará para el examen; el candidato debe entender los conceptos y ser capaz de proporcionar ejemplos.*

Inglés	Español
A/B testing	Pruebas A/B
Acceptance tests	Pruebas de aceptación
Affinity ( <i>in DevOps</i> )	Afinidad
Agile	Agile
Anti-pattern	Antipatrón
Application Deployment	Despliegue de Aplicaciones
Artifact Management	Gestión de Artefactos
(Artifact) repository	Repositorio (de Artefactos)
ATAM	ATAM
Automated testing	Pruebas automatizadas
Automation	Automatización
Binary files	Archivos binarios
Blamelessness	Libre de culpa / sin culpa
Blue-Green deployment	Despliegues "azul-verde" (Blue-Green deployment)
Build Management	Gestión de Compilaciones (Build management)
Build-time	Fase de Compilación (build-time)
Canary releasing	Canary releasing
Capacity testing	Pruebas de capacidad
Change Management	Gestión de Cambios
Check-in	Check-in
Cloud Computing	Computación en la nube / Cloud
Collaboration ( <i>in DevOps</i> )	Colaboración ( <i>en DevOps</i> )
Commit (stage)	(Fase de) commit (commit stage)
Communication styles	Estilos de comunicación
Compact	Pacto (Compact)
Component (tests)	Pruebas de componentes
Configuration Management	Gestión de la Configuración / Gestión de Configuraciones
Containers	Contenedores
Continuous Delivery	Entrega Continua
Continuous Deployment	Despliegue Continuo
Continuous Integration	Integración Continua
Cycle time	Tiempo de Ciclo
Definition of Done ( <i>in Agile Scrum</i> )	Definición de Hecho (DoD)
Dependency	Dependencia
(Deployment) Pipeline	Flujo (de Despliegue) / Pipeline (de Despliegue)
Development Team	Equipo de desarrollo
DevOps Engineer	Ingeniero DevOps (DevOps Engineer)
Disciplined Agile	Disciplined Agile
Distributed Team	Equipos de distribuidos
Effective DevOps	DevOps eficaz
Event Management	Gestión de Eventos
Exploratory testing	Pruebas exploratorias

Flow	Flujo
Functional acceptance tests	Tests / Pruebas de Aceptación Funcional
Forensic tools	Herramientas de análisis forense
Gatekeeper	Gatekeeper
Happy path	Happy path
Human error	Error Humano
Incident Management	Gestión de Incidencias
Information radiators	Radiadores de información
Infrastructure Automation	Automatización de Infraestructuras
Infrastructure Management	Gestión de Infraestructuras
INVEST	INVEST
Integration tests	Pruebas de integración
Iteration	Iteración
ITSM (IT Service Management)	ITSM / Gestión de Servicios TI
Ji-Kotei-Kanketsu (JKK)	Ji-Kotei-Kanketsu (JKK)
Just-in-Time (JiT)	Just-in-time (JiT)
Kaizen ( <i>in Lean</i> )	Kaizen ( <i>en Lean</i> )
Lean	Lean
Libraries	Bibliotecas
Light-weight ITSM	Gestión simplificada de Servicios TI (light-weight ITSM)
Manual testing	Pruebas manuales
Minimum Viable Product	Mínimo Producto Viable / Producto Mínimo Viable
Monitoring strategy	Estrategia de monitorización
Negotiation styles	Estilos de negociación
Non-functional testing	Pruebas no funcionales
Obeya	Obeya
Orchestration	Orquestación
One-piece-flow	Flujo de una pieza
Operations Team	Equipo de operaciones
Organizational Learning	Aprendizaje Organizacional
Plan-Do-Check-Act cycle (PDCA cycle)	Ciclo PDCA / Plan-Do-Check-Act
Process Master	Process Master
Product Backlog	Backlog de producto
Product Owner ( <i>in Agile Scrum</i> )	Product Owner / Propietario de Producto ( <i>en Agile Scrum</i> )
Project charter	Acta de constitución del proyecto ( <i>Project charter</i> )
Pull system	Sistema pull
Quality Assurance (QA)	Aseguramiento de la Calidad
Regression testing	Pruebas de regresión
Release Coordinator	El Coordinador de versiones
Reliability Engineer	Ingeniero de fiabilidad
Retrospective	Retrospectiva
Rhythm ( <i>in Lean</i> )	Ritmo ( <i>en Lean</i> )
Root Cause Analysis	Análisis de Causa Raíz
Run time	Tiempo de Ejecución
Sad path	Sad path
Scaling ( <i>of DevOps or Agile Scrum</i> )	Escalado ( <i>de DevOps o de Agile Scrum</i> )
Scrum	Scrum
Scrum Master ( <i>in Agile Scrum</i> )	Scrum Master ( <i>en Agile Scrum</i> )
Service Continuity	Continuidad del Servicio
Service Level Agreement (SLA)	Acuerdo de Nivel de Servicio (SLA / ANS)
Service Master	Service Master
Silos	Silos

Sprint	Sprint
System of Engagement (SoE)	System of Engagement (SoE)
System of Record (SoR)	System of Record (SoR)
System tests	Pruebas de sistema
Test-Driven Development	Desarrollo Dirigido por Pruebas (Test-Driven Development, TDD)
Tools	Herramientas
Toyota Production System (TPS)	Sistema de Producción de Toyota (TPS)
Unit Test	Prueba unitaria
Usability tests	Pruebas de usabilidad
User Acceptance Testing (UAT)	Pruebas de Aceptación de Usuario (UAT)
User Story	Historia de Usuario
Value Stream Mapping	Mapa de la Cadena de Valor (VSM)
Velocity ( <i>in Agile Scrum</i> )	Velocidad ( <i>en Agile Scrum</i> )
(Vendor) lock-in	Dependencia (del proveedor)
Version Control	Control de Versiones
Virtualization	Virtualización
Waste ( <i>in Lean</i> )	Desperdicio (waste) ( <i>in Lean</i> )
Waterfall	Waterfall / cascada
Work-in-Progress (WiP)	Work in Progress (WiP)

## 4. Bibliografía

### Bibliografía para el examen

En la bibliografía siguiente figuran los conocimientos necesarios para el examen EXIN DevOps Master™.

- A. Jennifer Davis, Katherine Daniels  
**AEffective DevOps: Building a Culture of Collaboration, Affinity, and Tooling at Scale**  
O'Reilly Media; 1ª edición (el 25 de junio de 2016)  
ISBN-13: 978-1491926307  
ISBN-10: 1491926309
- B. Jez Humble, David Farley  
**Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation**  
Addison-Wesley Professional; 1ª edición (el 6 de agosto de 2010)  
ISBN-13: 978-0321601919  
ISBN-10: 0321601912
- C. Koichiro (Luke) Toda, President Strategic Staff Services Corporation and Director of TPS Certificate Institution, and Nobuyuki Mitsui, CTO of Strategic Staff Services Corporation  
**Success with Enterprise DevOps**  
White Paper; junio de 2016 (se puede descargar de la página de producto de EXIN DevOps Master™)

### Bibliographie complémentaire

- D. Gene Kim, Kevin Behr, George Spafford  
**The Phoenix Project**  
IT Revolution Press (el 10 de enero de 2013)  
ISBN-10: 0988262576  
ISBN-13: 978-0988262577
- E. Gene Kim, Jez Humble, Patrick Debois, John Willis  
**The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations**  
IT Revolution Press; 1ª edición (2016)  
ISBN-10: 1942788002  
ISBN-13: 978-1942788003
- F. Otros recursos:  
<http://newrelic.com/devops>  
<http://devops.com/>

### Explicación

La bibliografía adicional es solo para consultar y profundizar conocimientos. Se recomienda encarecidamente leer "The Phoenix Project" antes de la formación.

## Matriz Bibliográfica

Requisito de examen		Bibliografía
<b>1. Adopción de DevOps</b>		
1.1	La Mentalidad DevOps y los Beneficios	
1.1.1	analizar antipatronos DevOps en un escenario dado	A: Capítulo 5 B: Capítulo 1 C: Capítulo 2
1.1.2	explicar los beneficios de DevOps	B: Capítulo 1 C: Capítulo 4
1.1.3	explicar por qué DevOps encaja tan bien en los procesos de desarrollo de software actuales	A: Capítulo 3 B: Capítulo 1 C: Capítulo 4
1.1.4	explicar por qué DevOps requiere de una forma especial de pensamiento para funcionar	A: Capítulo 1, 2, 4 C: Capítulo 4
1.1.5	explicar cómo encaja DevOps con las prácticas Lean y Agile Scrum	B: Capítulo 15 C: Capítulo 1, 4
1.2	Cultura Organizativa	
1.2.1	explicar la importancia de los cuatro pilares para un DevOps efectivo (Colaboración, Afinidad, Herramientas y Escalado)	A: Capítulo 6, 7
1.2.2	analizar un escenario en busca de carencias en alguno de los elementos que componen la mentalidad DevOps	A: Capítulo 7
1.2.3	explicar cómo crear un equipo a partir de un grupo de personas fomentando la colaboración, la mentalidad DevOps, la empatía y la confianza	A: Capítulo 7, 9
1.2.4	analizar una situación en la que haya ideas erróneas sobre la colaboración y sugerir métodos que resuelvan el problema	A: Capítulo 7, 8, 9, 16
1.2.5	analizar una situación en la que exista la necesidad de gestión de conflictos y plantear la mejor solución	A: Capítulo 7, 14
1.2.6	explicar cómo la gestión de recursos humanos puede fomentar la diversidad y el impacto beneficioso de ésta sobre la organización	A: Capítulo 7
1.3	Principios y Conceptos	
1.3.1	explicar el uso y la utilidad de las diferentes metodologías de desarrollo de software (Waterfall, Agile, Scrum) y sus principios básicos	A: Capítulo 4 B: Capítulo 1 C: Capítulo 5
1.3.2	explicar el uso y la utilidad de las diferentes metodologías utilizadas en operaciones (IT Service Management (Gestión de Servicios TI))	C: Capítulo 3, 4
1.3.3	explicar el uso y la utilidad de los métodos Lean	A: Capítulo 4
<b>2. Planificación, requerimientos y diseño</b>		
2.1	Gestión del Ciclo de vida de Aplicaciones o Servicios	
2.1.1	explicar cómo DevOps añade valor a la Gestión moderna del Ciclo de Vida de las Aplicaciones	C: Capítulo 1
2.1.2	explicar cómo DevOps mejora la experiencia del cliente cuando se usa para la Gestión del Ciclo de Vida del Servicio	C: Capítulo 4



	2.2	Plan de Proyecto (Defining Scope) y Gestión Visual	
	2.2.1	explicar cómo se determina el ámbito de un proyecto DevOps	B: Capítulo 10 C: Capítulo 8
	2.2.2	explicar por qué la Gestión Visual de un proyecto DevOps facilita las prácticas DevOps	B: Capítulo 5 C: Capítulo 7
	2.3	Diseño de Infraestructuras y Arquitectura	
	2.3.1	explicar cómo DevOps cambia o influye en el diseño de infraestructuras y arquitecturas TI	B: Capítulo 11
	2.3.2	explicar por qué Computación en la nube y técnicas de virtualización facilitan la incorporación de Dev y Ops	B: Capítulo 11
	2.4	Acuerdos y Requerimientos de Nivel de Servicio	
	2.4.1	explicar cómo DevOps cambia los Requisitos de Nivel de Servicio (SLR) y los Acuerdos de Nivel de Servicio (SLA)	B: Capítulo 12 C: Capítulo 4
	2.5	Implementación de una Estrategia de Pruebas	
	2.5.1	explicar por qué y cómo debe ser modificada la Estrategia de Pruebas cuando se hace la transición a DevOps	B: Capítulo 4
	2.5.2	analizar y comprobar la integridad de las Historias de Usuario	B: Capítulo 4
<b>3. Desarrollo y Despliegue</b>			
	3.1	Entrega e Integración Continuas	
	3.1.1	explicar por qué la Entrega Continua es esencial para un DevOps efectivo	B: Capítulo 13, 15
	3.1.2	analizar cómo integrar la Entrega Continua en un escenario	B: Capítulo 3, 5, 15
	3.1.3	analizar cómo resolver los problemas de Entrega Continua en un escenario	B: Capítulo 15
	3.1.4	explicar por qué la Integración Continua es esencial para un DevOps efectivo	B: Capítulo 3
	3.1.5	analizar cómo conseguir la Integración Continua en un escenario, con equipo distribuido o con un control de versiones distribuido	B: Capítulo 3
	3.1.6	analizar cómo resolver problemas en un escenario de Integración Continua	B: Capítulo 3
	3.2	Pipeline de Despliegue	
	3.2.1	explicar la lógica de la anatomía de un pipeline de despliegue	B: Capítulo 5, 8 C: Capítulo 7
	3.2.2	explicar cómo utilizar técnicas de <i>scripting</i> para la construcción ( <i>build</i> ) y el despliegue	B: Capítulo 1, 6
	3.3	Despliegue Continuo	
	3.3.1	explicar cómo se deben modificar el plan de versiones y de iteraciones para adaptarse a DevOps	C: Capítulo 5, 7
	3.3.2	analizar cómo implementar el Despliegue Continuo en un escenario	B: Capítulo 10

3.4	Ji-Kotei-Kanketsu, Ritmo, Work-in-Progress (WIP) y Flujo de una pieza	
3.4.1	explicar los conceptos <i>Ji-Kotei-Kanketsu</i> , Ritmo, <i>Work-in-Progress</i> (WIP) y Flujo de una pieza	C: Capítulo 4, 7
3.4.2	analizar un escenario buscando un problema al utilizar los conceptos de <i>Ji-Kotei-Kanketsu</i> , Ritmo, <i>Work-in-Progress</i> (WIP) y Flujo de una pieza y encontrar una solución adecuada	A: Capítulo 1, 2, 9 B: Capítulo 1 C: Capítulo 4
3.5	Automatización, Herramientas y Pruebas (Testing)	
3.5.1	explicar por qué la automatización es importante para un DevOps efectivo	B: Capítulo 1, 8
3.5.2	explicar cómo utilizar las herramientas para facilitar DevOps en general	B: Capítulo 8
3.5.3	explicar cómo utilizar las herramientas para apoyar la mentalidad y la cultura DevOps	A: Capítulo 12
3.5.4	explicar por qué es fundamental que se automaticen las pruebas	A: Capítulo 1 B: Capítulo 3
3.5.5	analizar un escenario y escoger la forma correcta de automatizar un test de aceptación	B: Capítulo 8
<b>4. Operación y Escalado</b>		
4.1	Gestión de Datos; Infraestructuras, y Entornos, y Componentes y Dependencias	
4.1.1	explicar los problemas que se encuentran al manejar datos de bases de datos en entornos DevOps	B: Capítulo 12
4.1.2	analizar un escenario donde se utiliza una base de datos en un entorno DevOps y dar la mejor solución al problema	B: Capítulo 12
4.1.3	analizar un escenario e identificar la mejor forma de preparar la infraestructura para el despliegue, o bien para gestionarla después del despliegue	B: Capítulo 11
4.1.4	analizar un escenario y sugerir estrategias comunes para la gestión de componentes	B: Capítulo 13
4.1.5	explicar cómo gestionar dependencias	B: Capítulo 13
4.2	Gestión de Configuraciones y Control de Versiones	
4.2.1	explicar por qué el control de versiones es clave para un DevOps efectivo	B: Capítulo 2
4.2.2	explicar cómo mantener un control de versiones sobre datos, infraestructuras y componentes	B: Capítulo 10, 11
4.2.3	analizar un escenario y sugerir la mejor estrategia para atacar un problema de gestión de configuraciones	B: Capítulo 2
4.3	Cloud e Infraestructuras Estáticas	
4.3.1	explicar cuándo es o no es necesario migrar a una infraestructura basada en la nube para un DevOps efectivo	A: Capítulo 17 B: Capítulo 11 C: Capítulo 5, 7
4.3.2	explicar cómo se debe gestionar la infraestructura basada en la nube en un entorno DevOps	B: Capítulo 11

	4.4	Continuidad de Negocio	
	4.4.1	explicar cómo DevOps facilita las prácticas de Continuidad del Negocio	C: Capítulo 4
	4.5	Escalado	
	4.5.1	analizar un escenario y explicar si es necesario realizar un escalado incremental o una reducción ( <i>scale-up or scale-down</i> ) en esa situación y por qué e identificar la mejor forma de hacerlo.	A: Capítulo 14
	4.5.2	analizar un escenario buscando qué ha ido mal en una situación de escalado e identificar una manera adecuada de resolver el problema	A: Capítulo 15 B: Capítulo 11
	4.5.3	explicar cómo las políticas sociales y de contratación apoyan el escalado de DevOps	A: Capítulo 6, 7, 14
<b>5. Fin de la Vida Útil</b>			
	5.1	Condiciones de Fin de la Vida Útil para un producto o servicio	
	5.1.1	explicar qué condiciones se deben satisfacer antes de terminar un producto o servicio.	C: Capítulo 7

Nota: La lectura del recurso bibliográfico D, "The Phoenix Project" resultará especialmente beneficiosa para la comprensión de las siguientes especificaciones:

- 1.1
- 1.2
- 3.1
- 3.3
- 3.4
- 4.4

# Contacto EXIN

[www.exin.com](http://www.exin.com)

