

Architecture Specialization

Examen de Muestra

Antes de Empezar

Este examen de muestra tiene 10 preguntas que le ayudarán a prepararse para el examen Architecture Specialization.

- Le recomendamos que prepare un entorno de examen tan real como sea posible.
- Reserve una habitación tranquila solo para usted.
- Imprima este documento, además de la última página.
- Consiga un cronómetro o configure un temporizador para la duración (recomendada) de 40 minutos.

La última página de este documento tiene las respuestas correctas. ¡No la vea! Úsela sólo después de completar su examen, para verificar qué tan bien lo hizo.

Durante el Examen Muestra

Para simular con precisión el ambiente real del examen, le sugerimos que:

- Lea atentamente cada pregunta y sus respuestas.
- ¡Tome su tiempo! Puede volver a revisar las preguntas y puede cambiar sus elecciones.
- Marque las preguntas que desea revisar al final.
- Elija solo una respuesta por pregunta, ya que solo una es la correcta.
- Responda todas las preguntas, puesto que no hay ningún beneficio en no hacerlo.
- Intente apagar todos los dispositivos electrónicos durante el examen.
- Absténgase de usar o leer cualquier material externo durante el examen.

Después de completar el Examen Muestra

Después de completar el examen, valide las respuestas que seleccionó revisando las que se proporcionan en la última página de este documento y cuente el número total de respuestas correctas. Debido a que el puntaje para aprobar es del 70 % o más debe responder correctamente al menos 7 preguntas. En caso de que elija alguna respuesta incorrecta, le sugerimos que revise los materiales de estudio donde se cubre ese tema específico.

Preguntas del examen muestra

1. ¿Cuál de las siguientes opciones describe un beneficio de adoptar el Architecture Canvas?

- ☐ **A.** El Architecture Canvas proporciona un enfoque sistemático para el diseño de la arquitectura, respaldado por una herramienta de validación.
 - ☐ **B.** El Architecture Canvas proporciona una forma automática de seguir y corregir los principios de la arquitectura.
 - ☐ **C.** El Architecture Canvas proporciona una forma de promover con facilidad la colaboración y la comprensión de los usuarios de negocio.
 - ☐ **D.** El Architecture Canvas proporciona un diseño de la arquitectura más rápido sin necesidad de validación.
-

2. En OutSystems, ¿se recomienda unir conceptos no relacionados en el mismo módulo?

- ☐ **A.** Sí, porque reduce el número de referencias, lo que simplifica las implementaciones.
 - ☐ **B.** No, porque evita la independencia del ciclo de vida de los conceptos y agrega efectos innecesarios a los consumidores.
 - ☐ **C.** Sí, para evitar referencias cíclicas y reducir la huella del código generado.
 - ☐ **D.** No, porque agrega complejidad innecesaria a los consumidores que requieren ambos conceptos.
-

3. Considere la siguiente afirmación: *“En OutSystems 11, se recomienda utilizar Screens relacionadas con el negocio en los módulos de la capa Foundation, debido a que no infringe ninguna regla de validación de la arquitectura.”* ¿Cuál de las siguientes opciones es **correcta**?

- ☐ A. La oración es verdadera para todas las Screens relacionadas con el negocio. Todas las referencias a las Screens son débiles, por lo que pueden y deben usarse en los módulos de la capa Foundation, sin ocasionar referencias ascendentes.
 - ☐ B. La oración es falsa. Aunque todas las referencias a las Screens son débiles, solo las pantallas no relacionadas con el negocio deben usarse en los módulos de la capa Foundation.
 - ☐ C. La oración es verdadera para las Screens web relacionadas con el negocio. Las referencias a las Screens móviles todavía se consideran referencias sólidas, por lo que no deben usarse en los módulos de la capa Foundation.
 - ☐ D. La oración es falsa. Aunque todas las referencias a las Screens son débiles, solo las Screens relacionadas con el negocio que no están consumiendo otros módulos deben usarse en los módulos de la capa Foundation.
-

4. Considere que tenemos una referencia lateral débil entre módulos End-user. ¿Cuál es la mejor manera de eliminar esa referencia para evitar una infracción de la arquitectura?

- ☐ A. Reemplazar los destinos de la pantalla por URL externas y acciones de consumo a través de la API de REST.
 - ☐ B. Se permiten referencias laterales débiles entre módulos End-user, por lo que no es necesario eliminar esa referencia.
 - ☐ C. En estos escenarios, mueva siempre los elementos consumidos del módulo productor al módulo consumidor para evitar la referencia.
 - ☐ D. Identificar los elementos consumidos, sin incluir los destinos, y moverlos a módulos Core / Foundation reutilizables.
-

5. ¿Cuál de las siguientes opciones describe una razón válida para crear un módulo Calculation Engine (*_Eng*)?

- ☐ A. Abstraer diferentes lógicas complejas implementadas en varios módulos Business Logic.
 - ☐ B. Admitir un concepto con entidades y edición CRUD.
 - ☐ C. Admitir cálculos complejos (por ejemplo, un simulador de seguros).
 - ☐ D. Admitir acciones reutilizables de los módulos Core Widget.
-

6. ¿Por qué es tan importante adoptar una convención de nomenclatura en nuestras aplicaciones?

- ☐ A. Para revelar la naturaleza de cada módulo.
 - ☐ B. Para hacer cumplir la arquitectura de referencia.
 - ☐ C. Para normalizar patrones.
 - ☐ D. ¡Todas las opciones restantes son correctas!
-

7. Al mover elementos entre módulos en OutSystems, ¿a qué elementos debemos prestar más atención y actuar con más cuidado?

- ☐ A. A los elementos sin persistencia de datos, como acciones, estructuras o bloques. Debido a que estos elementos no son persistentes, pueden eliminarse de manera involuntaria en el proceso.
 - ☐ B. A los elementos con persistencia de datos, como las entidades. Al mover una entidad, solo se mueve su definición lógica. Físicamente, es otra tabla de base de datos.
 - ☐ C. No hay ningún elemento en particular. Todos los elementos que se mueven entre módulos mantienen su integridad.
 - ☐ D. Solo los elementos configurables, como temporizadores, propiedades del sitio o roles. Estos elementos son los únicos que tienen una persistencia de datos que difiere de desarrollo a producción.
-

8. Considere la siguiente captura de pantalla del Discovery y las evidencias de problemas identificadas. ¿Cuál de estas evidencias debe resolverse primero?

Enduser			
Customer Orders 1 → 1	Customer Orders Mobile 1 → 1	DirectoryApp 5 → 5	Field Services BO 12 → 12
Field Services Mobile 1 → 21	Orders Mobile 1 → 1	OSMDB 1 → 1	OSMDB Mobile 1 → 1
Recruitment 3 → 3	Time Management 1 → 3	Vacations Management 1 → 2	
Core			
Employee Services 5 → 4	OutSystems AI Language Anal... 1 → 1		
Foundation			
Ext Auth Style Guide 2 → 2	OutSystems UI Mobile Style ... 1 → 1	OutSystems UI Reactive Styl... 1 → 1	SQL Server Connector 2 → 1

- ☐ A. La pequeña proporción de Core applications muestra que no hay suficiente capacidad de reutilización de los módulos Core.
- ☐ B. La única forma de resolver todas las referencias ascendentes es refactorizando los módulos dentro de cada aplicación.
- ☐ C. Hay demasiados servicios compartidos dentro de los módulos End-User.
- ☐ D. No se pueden mover módulos entre aplicaciones.

9. ¿Por qué es tan importante tener una infraestructura de varias capas para admitir el diseño de la arquitectura?

- ☐ A. Para promover una abstracción correcta de los servicios reutilizables.
- ☐ B. Para optimizar la independencia del ciclo de vida.
- ☐ C. Para minimizar el efecto de los cambios.
- ☐ D. ¡Todas las opciones restantes son correctas!

10. ¿En cuál de los siguientes escenarios se debería implementar una Style Guide (guía de estilo) personalizada?

- ☐ **A.** Siempre que las aplicaciones no se puedan beneficiar de ningún tema existente.
La guía de estilo se debe implementar clonando la Screen más compleja disponible, para aprovechar los Blocks y Screen Actions de la Screen original y acelerar las implementaciones.
- ☐ **B.** Siempre que las aplicaciones no se puedan beneficiar de ningún tema existente.
La guía de estilo se debe implementar partiendo de la Screen más básica posible, debido a que solo contiene la estructura mínima requerida para comenzar.
- ☐ **C.** Siempre que una aplicación necesite tener implementaciones independientes y bajo nivel de interdependencia entre otras aplicaciones.
- ☐ **D.** Siempre. Cada proyecto / aplicación debe tener su propia guía de estilo personalizada.

Respuestas

1. A
2. B
3. B
4. B
5. C
6. D
7. B
8. C
9. D
10. B