

Examen de QA: Automatización con Katalon Studio

Duración: 3 horas | Sitio de Prueba: <https://shop.geekqa.net/>

FASE 1: CONFIGURACIÓN INICIAL (30 minutos)

1.1 Configuración de JIRA (15 minutos)

Instrucciones: 1. Crear un nuevo proyecto en JIRA con la clave "SQAT" (Shop QA Testing) 2. Configurar los siguientes tipos de issues: - Epic - Story - Task - Bug - Test Case

3. Crear el Epic principal:

- **Título:** "Automatización de Pruebas E-commerce Shop GeekQA"
- **Descripción:** "Implementar suite de pruebas automatizadas para validar funcionalidades críticas del e-commerce"

1.2 Configuración de Katalon Studio (15 minutos)

Instrucciones: 1. Abrir Katalon Studio 2. Crear nuevo proyecto: "ShopGeekQA_Automation" 3. Configurar el navegador Chrome como default 4. Verificar que el sitio <https://shop.geekqa.net/> sea accesible

FASE 2: ANÁLISIS Y HISTORIAS DE USUARIO (45 minutos)

2.1 Exploración del Sitio (15 minutos)

Tareas: 1. Explorar todas las funcionalidades disponibles en el sitio 2. Documentar en JIRA como comentarios del Epic: - Funcionalidades encontradas - Flujos principales identificados - Elementos críticos para testing

2.2 Creación de Historias de Usuario en JIRA (30 minutos)

Crear las siguientes User Stories en JIRA (vincular al Epic creado):

US-001: Navegación por Catálogo de Productos

- **Como** usuario del e-commerce
- **Quiero** navegar por el catálogo de productos
- **Para** encontrar artículos de mi interés

Criterios de Aceptación: - Ver lista de productos en la página principal - Poder hacer clic en un producto específico - Visualizar detalles del producto seleccionado

DoD (Definition of Done): - Prueba manual documentada - Test automatizado creado - Test ejecutado exitosamente

US-002: Gestión del Carrito de Compras

- **Como** cliente potencial
- **Quiero** agregar productos al carrito
- **Para** gestionar mi selección de compra

Criterios de Aceptación: - Agregar productos al carrito desde la página de detalle - Visualizar productos agregados en el carrito - Modificar cantidades de productos - Eliminar productos del carrito - Ver total de la compra actualizado

US-003: Proceso de Búsqueda de Productos

- **Como** usuario
- **Quiero** buscar productos específicos
- **Para** encontrar rápidamente lo que necesito

Criterios de Aceptación: - Ingresar términos de búsqueda en campo de búsqueda - Obtener resultados relevantes - Manejar búsquedas sin resultados apropiadamente

US-004: Responsividad del Sitio

- **Como** usuario móvil
- **Quiero** que el sitio sea responsive
- **Para** tener una buena experiencia en cualquier dispositivo

Criterios de Aceptación: - El sitio se adapta correctamente a diferentes resoluciones - Elementos clickeables son accesibles en móvil - Navegación funciona en dispositivos táctiles

US-005: Validación de Formularios

- **Como** usuario
- **Quiero** recibir validaciones claras en formularios
- **Para** completar procesos sin errores

Criterios de Aceptación: - Validaciones de campos obligatorios - Mensajes de error claros y específicos - Validación de formato de datos (emails, teléfonos, etc.)

FASE 3: IMPLEMENTACIÓN DE PRUEBAS AUTOMATIZADAS (90 minutos)

3.1 Configuración Inicial en Katalon (10 minutos)

1. Crear carpetas de organización en Test Cases:
 - Navigation_Tests
 - Shopping_Cart_Tests
 - Search_Tests
 - Responsive_Tests
 - Form_Validation_Tests
2. Crear Object Repository con los siguientes elementos base:
 - HomePage
 - ProductPage
 - ShoppingCart
 - SearchResults

3.2 Test Case 1: Navegación por Catálogo (20 minutos)

Archivo: Navigation_Tests/TC001_Product_Catalog_Navigation

Pasos detallados:

1. Abrir navegador
2. Navegar a `https://shop.geekqa.net/`
3. Verificar que la página principal carga correctamente
4. Verificar que hay productos visibles
5. Hacer clic en el primer producto disponible
6. Verificar que se abre la página de detalle del producto
7. Verificar elementos presentes: título, precio, descripción, botón agregar
8. Regresar a página principal
9. Cerrar navegador

Objetos necesarios en Object Repository: - `btn_FirstProduct` (xpath: primer producto en la lista) - `lbl_ProductTitle` (título del producto en página detalle) - `lbl_ProductPrice` (precio del producto) - `btn_AddToCart` (botón agregar al carrito)

Vincular en JIRA: - Crear Task “Automatizar TC001” vinculado a US-001 - Estado inicial: “To Do”

3.3 Test Case 2: Funcionalidad del Carrito (25 minutos)

Archivo: Shopping_Cart_Tests/TC002_Add_Product_To_Cart

Pasos detallados:

1. Abrir navegador y navegar al sitio
2. Seleccionar un producto específico
3. Hacer clic en "Agregar al carrito"

4. Verificar mensaje de confirmación o redirección
5. Navegar al carrito de compras
6. Verificar que el producto aparece en el carrito
7. Verificar cantidad correcta
8. Verificar precio correcto
9. Modificar cantidad del producto
10. Verificar actualización del total
11. Eliminar producto del carrito
12. Verificar que carrito queda vacío

Variables de Test Data: - product_name = “Nombre del producto a probar”
- initial_quantity = 1 - modified_quantity = 3

Vincular en JIRA: - Crear Task “Automatizar TC002” vinculado a US-002

3.4 Test Case 3: Funcionalidad de Búsqueda (20 minutos)

Archivo: Search_Tests/TC003_Product_Search

Pasos detallados:

1. Abrir navegador y navegar al sitio
2. Localizar campo de búsqueda
3. Ingresar término de búsqueda válido
4. Ejecutar búsqueda
5. Verificar que se muestran resultados
6. Verificar que resultados son relevantes al término buscado
7. Probar búsqueda con término inexistente
8. Verificar manejo apropiado de "sin resultados"
9. Probar búsqueda con caracteres especiales
10. Verificar que no hay errores en la aplicación

Test Data: - valid_search_term = “duck” (o término relevante según productos) - invalid_search_term = “xyzabc123” - special_characters = “@#\$\$%”

3.5 Test Case 4: Responsive Design (15 minutos)

Archivo: Responsive_Tests/TC004_Mobile_Responsiveness

Pasos detallados:

1. Configurar navegador con resolución desktop (1920x1080)
2. Navegar al sitio y verificar layout
3. Cambiar resolución a tablet (768x1024)
4. Verificar que elementos se adaptan correctamente
5. Cambiar resolución a móvil (375x667)
6. Verificar navegación mobile
7. Probar funcionalidad de tap en elementos principales
8. Verificar que texto es legible
9. Verificar que botones son accesibles

Usar WebDriver commands: - WebUI.setViewportSize(width, height)
- WebUI.verifyElementVisible() - WebUI.verifyElementClickable()

FASE 4: EJECUCIÓN Y REPORTE (30 minutos)

4.1 Ejecución de Test Suite (15 minutos)

Instrucciones: 1. Crear Test Suite “ShopGeekQA_Complete_Suite” 2. Incluir todos los test cases creados 3. Configurar para ejecución en Chrome 4. Ejecutar suite completa 5. Capturar screenshots de fallos (si los hay)

4.2 Documentación de Resultados en JIRA (15 minutos)

Para cada Test Case: 1. Actualizar status del Task a “Done” si pasa 2. Si falla, crear Bug con: - Título descriptivo - Pasos para reproducir - Resultado esperado vs actual - Screenshot adjunto - Severidad y prioridad

Formato de Bug Report:

Título: [TC00X] Descripción breve del error

Tipo: Bug

Prioridad: Alta/Media/Baja

Componente: Área afectada

Descripción:

Pasos para reproducir:

- 1.
- 2.
- 3.

Resultado Esperado:

Resultado Actual:

Evidencia: [Adjuntar screenshot]

FASE 5: ANÁLISIS Y MEJORAS (15 minutos)

5.1 Análisis de Cobertura

Documentar en JIRA como comentario del Epic: 1. Porcentaje de pruebas automatizadas exitosas 2. Funcionalidades cubiertas vs no cubiertas 3. Riesgos identificados 4. Recomendaciones para siguientes iteraciones

5.2 Plan de Mejora

Crear Task en JIRA: “Plan de Mejora de Automation” - Identificar 3 mejoras técnicas para los scripts - Sugerir 2 casos de prueba adicionales -

Proponer estrategia de mantenimiento

CRITERIOS DE EVALUACIÓN

Configuración de JIRA (20 puntos)

- ☒ Proyecto correctamente configurado (5 pts)
- ☒ Epic y User Stories bien estructuradas (10 pts)
- ☒ Tasks vinculados apropiadamente (5 pts)

Implementación Técnica (40 puntos)

- ☒ Test Cases funcionalmente correctos (20 pts)
- ☒ Object Repository bien organizado (10 pts)
- ☒ Uso apropiado de variables y data (10 pts)

Ejecución y Documentación (25 puntos)

- ☒ Suite ejecutada completamente (10 pts)
- ☒ Resultados documentados en JIRA (10 pts)
- ☒ Bugs reportados correctamente (5 pts)

Análisis Profesional (15 puntos)

- ☒ Análisis de cobertura realista (8 pts)
 - ☒ Recomendaciones técnicamente viables (7 pts)
-

RECURSOS ADICIONALES

Comandos Katalon Útiles:

```
// Navegación
WebUI.openBrowser('')
WebUI.navigateToUrl('URL')
WebUI.closeBrowser()

// Interacciones
WebUI.click(findTestObject('Object'))
WebUI.setText(findTestObject('Object'), 'text')
WebUI.selectOptionByValue(findTestObject('Object'), 'value', false)

// Verificaciones
WebUI.verifyElementPresent(findTestObject('Object'), timeout)
WebUI.verifyElementText(findTestObject('Object'), 'expected_text')
WebUI.verifyElementVisible(findTestObject('Object'))
```

```
// Screenshots  
WebUI.takeScreenshot()
```

Tips para Object Repository:

- Usar IDs cuando estén disponibles
- XPath como segunda opción
- CSS Selectors para elementos dinámicos
- Siempre incluir timeouts apropiados

¡Éxito en tu examen! Recuerda documentar todo en JIRA y seguir las mejores prácticas de QA.