


QA MANUAL SENIOR · PROCESO DE SELECCIÓN

Prueba Técnica

QA Manual Senior

 2 horas 30 minutos Web + App Mobile Entregable: documento saucedemo.com

S1 · Riesgos

S2 · Casos de Prueba

S3 · Ejecución & Bugs

S4 · Mobile / App

S5 · Regresión

S6 · Mejora

S7 · IA (opcional)

Entrega

✦ Antes de empezar — Consideraciones



Resuélvela tú, no la IA: sabemos que hoy la inteligencia artificial es una herramienta poderosa y de gran ayuda en el día a día. Sin embargo, esta prueba evalúa tu criterio, tu estrategia y las experiencias que has vivido en proyectos reales — eso una IA no lo puede responder por ti. Lo que nos interesa es conocer tu forma de pensar y tu recorrido como profesional, así que resuélvela con tus propias palabras.

Ahora bien — usar IA en tu trabajo es muy distinto y lo valoramos: si tienes experiencia en automatización con IA o has venido incorporándola en tus procesos de QA, eso suma mucho. Cuéntanoslo con ejemplos en la **Sección 7**, que creamos justo para eso.



Tiempo: la prueba está diseñada para tomar aproximadamente 2 horas 30 minutos. No es un examen contra reloj; tómate el tiempo que necesites con calidad.



Entregable: descarga el PDF o el Word desde los botones de arriba, complétalo y envíalo. También puedes usar Notion, Google Docs o el formato que prefieras.



Esta página es solo de lectura: aquí ves las instrucciones, pero las respuestas las escribes en el documento que descargues, no en este sitio.



Criterio sobre cantidad: valoramos cómo piensas y justificas tus decisiones más que el volumen. Calidad sobre cantidad.



Casos en Gherkin: los casos de prueba deben escribirse con la estructura Given / When / Then. Si usas otra metodología, mira la instrucción en la Sección 2.



Credenciales: usa las credenciales públicas de prueba que están más abajo. Para la app nativa, crea un usuario ficticio. No uses datos reales ni sensibles.



Documenta tus supuestos: si algo no queda claro o tomas una decisión, escríbelo. Eso también se evalúa.



¿Dudas? Si tienes alguna pregunta antes o durante la prueba, responde el correo con el que recibiste este enlace.

CONTEXTO

Eres parte del equipo de QA de una startup de e-commerce. El equipo actualmente es el **cuello de botella** del ciclo de releases: cada deploy espera validación manual y los sprints se acumulan.

Tu misión en esta prueba es doble: **demostrar tu capacidad técnica como QA senior** sobre una app real, y **proponer cómo el equipo puede dejar de ser ese cuello de botella** sin sacrificar calidad.



CREDENCIALES DE ACCESO



[Abrir saucedemo.com](https://saucedemo.com)

Usuario

Contraseña

Observación

standard_user

secret_sauce

Normal

Usuario	Contraseña	Observación
problem_user	secret_sauce	Comportamientos anómalos
performance_glitch_user	secret_sauce	Latencia inusual
locked_out_user	secret_sauce	No puede ingresar

1 Análisis de Riesgos

≈ 20 min

1.1 Mapa de riesgo

Analiza la aplicación y completa la tabla identificando las áreas de mayor riesgo. Justifica tu criterio.

Área / Funcionalidad	Nivel de riesgo	Justificación

1.2 Priorización

Dado que solo tienes **45 minutos** para ejecutar pruebas antes de un release urgente, ¿qué pruebas primero y qué dejas para después?

2.1 Flujo de compra completo

Diseña los casos de prueba para el flujo: **Login** → **Selección de producto** → **Carrito** → **Checkout** → **Confirmación**.

Metodología requerida: Gherkin

Todos los casos deben estar escritos usando la estructura **Given / When / Then**. Puedes complementar con **And** y **But** donde sea necesario.

ID	Título	Prioridad	Given	When	Then	And / But
TC-01						

¿Usas otra metodología?

No hay problema — escribe el resto de los casos en el formato que usas habitualmente, pero incluye **un ejemplo adicional** con esa metodología y explica brevemente por qué prefieres aplicarla y en qué contextos la consideras más útil que Gherkin.

*Lo que se evalúa es **qué eliges cubrir y por qué**, no la cantidad de casos.*

2.2 Casos negativos y de excepción

Lista al menos **5 casos negativos o de excepción** que consideres críticos. Escríbelos también en Gherkin.

ID	Título	Given	When	Then	Por qué es crítico
CN-01					
CN-02					
CN-03					
CN-04					

ID	Título	Given	When	Then	Por qué es crítico
CN-05					

3 Ejecución y Reporte de Bugs

≈ 30 min

3.1 Ejecución

Ejecuta pruebas sobre la aplicación web usando los usuarios disponibles. Presta especial atención a `problem_user` y `performance_glitch_user`.

3.2 Reporte de bugs

Reporta **todos los defectos** que encuentres. Mínimo 3 bugs esperados.

BUG-001

Título: _

Severidad: *Crítica / Alta / Media / Baja*

Prioridad: *Alta / Media / Baja*

Ambiente: *Web desktop / Mobile web / Navegador:*

Usuario: _

Precondiciones: _

Pasos: 1. 2. 3.

Resultado actual: _

Resultado esperado: _

Evidencia: *screenshot / video / log*

Repite el bloque para cada bug encontrado.

Esta sección tiene **dos partes**: mobile web y app nativa. No necesitas instalar nada especial.

Parte A — Mobile Web (responsive)

4.1 Comparativa web vs. mobile

Abre saucedemo.com desde el navegador de tu celular con el usuario `standard_user` y ejecuta el flujo **Login → Agregar producto → Checkout**.

Aspecto	¿Funciona igual que en desktop?	Diferencias observadas
Login		
Navegación del catálogo		
Agregar al carrito		
Formulario de checkout		
Confirmación de compra		

4.2 Bugs mobile

Si no encuentras bugs, documenta qué verificaste y por qué concluyes que no hay defectos. Eso también es válido.

Reporta usando el mismo formato de bug de la Sección 3.

📱 Parte B — App Nativa

Tú eliges la app, pero debe ser **robusta**: autenticación, flujos multi-paso, manejo de estado y al menos 2-3 módulos diferenciados (cuenta, catálogo, pagos, historial).

Ejemplos válidos: Mercado Libre, Amazon, Rappi, Pedidos Ya, Despegar, Airbnb, o similar.

⚠️ Nota sobre credenciales:

Estas apps requieren login. Puedes crear un usuario ficticio para la prueba, o si encuentras una app que permita modo invitado, también es válido. No uses credenciales reales ni datos personales sensibles.

Indica qué app elegiste, SO (iOS / Android), versión del SO y versión de la app.

4.3 Análisis rápido (10 minutos de exploración)

- ¿Cuáles son los 3 flujos más críticos para el negocio?
- ¿Qué área te generó más dudas o inconsistencias?
- ¿Detectaste algún bug? Repórtalo brevemente.

4.4 Casos específicos de mobile nativo

Diseña **5 casos de prueba** exclusivos o especialmente importantes en app nativa. Escríbelos en Gherkin.

ID	Título	Given	When	Then	Por qué es específico de mobile nativo
MOB-01					

ID	Título	Given	When	Then	Por qué es específico de mobile nativo
MOB-02					
MOB-03					
MOB-04					
MOB-05					

4.5 Escenario: el app "se va al fondo"

Estás probando el flujo de checkout en la app. En el paso 2 recibes una llamada, la atiendes y vuelves a la app 5 minutos después.

¿Qué esperas que pase? ¿Qué testeas? ¿Qué consideras un comportamiento aceptable vs. un bug?

5 Plan de Regresión

≈ 20 min

5.1 Estrategia de regresión

Diseña una estrategia de regresión **sostenible en el tiempo** para un equipo pequeño con releases cada dos semanas. Incluye:

- Qué cubriría tu suite de regresión
- Cómo la organizas (smoke, sanity, full regression)
- Cómo decides qué entra y qué sale con cada release

5.2 Regresión impactada

El equipo acaba de hacer un cambio en el módulo de **ordenamiento de productos** (sort by price / name). ¿Qué casos de regresión ejecutas y cuáles puedes saltar con confianza?

Esta sección no tiene respuestas incorrectas. Lo que nos interesa es tu forma de pensar el problema.

Situación:

El equipo de QA es el cuello de botella: cada feature espera validación manual antes de poder salir. Los developers ya terminaron sus cambios pero QA tarda 3-4 días en validar cada release. El negocio quiere deployar más rápido.

6.1 Diagnóstico

¿Cuáles crees que son las causas más probables? ¿Qué preguntas o investigas para confirmar tu hipótesis?

6.2 Propuesta concreta

Propone un plan para reducir el tiempo de validación **sin bajar la calidad**. Qué cambiarías primero, qué herramientas usarías, cómo medirías el éxito.

6.3 Herramientas y recursos

¿Qué herramientas o técnicas usas actualmente para hacer tu trabajo más eficiente? ¿Hay algo que hayas incorporado recientemente que cambió tu forma de trabajar?

Esta sección es **completamente opcional** y no resta puntos si no la completas. Pero si has trabajado con automatización o has incorporado IA en tus procesos de QA, este es el espacio para contarlo — y para nosotros es muy valioso conocerlo.

7.1 Tu experiencia

¿Has incorporado automatización o inteligencia artificial en tu trabajo como QA? Cuéntanos qué hiciste, en qué contexto y qué problema resolvía.

7.2 Herramientas

¿Qué herramientas usaste? (frameworks de automatización como Playwright, Cypress, Selenium, Appium; herramientas de IA como Copilot, ChatGPT, TestRigor, Applitools, etc.)

7.3 Caso de negocio o resultado

Si lo deseas, comparte un caso de negocio o un resultado concreto: por ejemplo, cuánto tiempo ahorró, cómo mejoró la cobertura o la calidad, o qué impacto tuvo en el equipo.

7.4 Evidencias (opcional)

Solo si quieres, adjunta enlaces, capturas, repositorios o cualquier material que respalde lo que contaste. No es obligatorio, pero suma.

ENTREGA

Cuando termines, envía **únicamente** estos dos elementos:

- El documento resuelto en **PDF** (adjunto)
- El **enlace a tu Google Doc**, con permiso de lectura activado

No aceptamos otros formatos (Word, Notion, Confluence, Markdown, etc.). Verifica que el enlace de Google Doc tenga permisos de visualización antes de enviarlo.

Si tomaste supuestos o algo no quedó claro, documéntalos dentro del documento — eso también se evalúa. Cualquier duda, responde este correo. ¡Mucha suerte!