

Módulo 2 / Encuentro 7/17

Plan de

prueba 1/2

## 

## 

## OBJETIVOS DEL MÓDULO 2

¿Qué habilidades desarrollarás?

* Manejo y priorización de la información
* Introducción al manejo de proyectos
* Análisis y diseño de pruebas

¿Qué herramientas técnicas aprenderás?

* Procesos de Prueba
* Análisis de Requerimientos
* Matriz de Trazabilidad de Requerimientos
* Historias de usuario
* Documentación de pruebas
* Introducción a formularios HTML



| Conceptos clave en el encuentro de hoy  * Procesos de Prueba * Análisis de Requerimientos * Matriz de Trazabilidad de Requerimientos * Análisis de Automatización * Tipos de requisitos |
| --- |

¡Ya hemos llegado al comienzo del módulo 2! A partir de ahora ya no tendrás marcado el uso de los tiempos ya que has adquirido progresivamente la suficiente autonomía como para administrarlo tú mismo. Por supuesto que podrás ir conversando con tu equipo para ver qué tal vienen a medida que vayan avanzando con el contenido de hoy.

Podrás notar que las guías comienzan a tener más páginas de aquí en adelante. ¡Ten calma! Ya eres un experto gestionando los tiempos. Además, encontrarás que a veces hacemos breves repasos de temas ya vistos y que los tópicos y ejercicios están acompañados de imágenes. Te proponemos no ver la cantidad de páginas si es algo que suele inquietarte un poco.

Hacia el final de esta guía, te encontrarás con un formulario que te llevará aproximadamente 20 minutos. Guárdate un espacio para completarlo. Este mismo formulario es un Check de conocimiento que te permitirá saber qué tal vienes hasta aquí y es el mismo que seguramente has visto antes de ingresar a esta guía.

# Introducción

Ya has tenido una introducción al mundo del testing. Así como has visto el ciclo de vida de producción de software, estaremos viendo las etapas del Ciclo de Vida del Testing (STLC). A partir de esta guía profundizaremos sobre cada etapa.

**MATERIAL DE LECTURA **

## Procesos de prueba

Como has estado aprendiendo, no existe un único proceso para realizar pruebas. Sin embargo, podemos tener en cuenta una serie de actividades que son comunes a todos estos procesos. Estas actividades ayudan a que el Testing alcance los objetivos establecidos.

Debemos tener en cuenta que un proceso de pruebas se ve afectado por una gran cantidad de factores internos y externos que deberemos tener en cuenta a la hora de planificar. Nombraremos algunos a continuación, pero deberás tener presente que ninguna lista es exhaustiva y que dependerá de cada proceso o entorno en que se aplique.

### Factores que pueden afectar el ciclo de pruebas:

* Modelo de Ciclo de vida del desarrollo del Software y metodologías del proyecto en uso[[1]](#footnote-0).
* Niveles y tipos de pruebas considerados.
* Riesgos del producto y del proyecto.
* Restricciones operativas del entorno (económicos, financieros, contractuales, temporales, etc).
* Políticas o estándares requeridos.

Teniendo esto en cuenta, sabremos que cada proceso de prueba será único y diferente del resto, pero dentro de estas actividades comunes, podemos identificar una serie de etapas a transitar en orden.

Pasos comunes a los procesos de prueba

1. Análisis de requerimientos
2. Planificación
3. Diseño y análisis
4. Implementación – Configuración del entorno
5. Ejecución
6. Actividades de cierre
7. Control y monitoreo

Iremos desarrollando estos pasos a lo largo de las siguientes guías.[[2]](#footnote-1)

## Análisis de requerimientos

Hasta el encuentro de hoy, estuvimos viendo los requerimientos como una categoría amplia en la que incluimos “todo lo que puede pedirse al momento de iniciar un proyecto de software.”

Ahora estamos en condiciones de ser más precisos y definir al requisito de software como **una descripción detallada del sistema en implementación**. Los requerimientos describen el **uso práctico del producto o servicio, la condición o la capacidad a la que debe ajustarse un sistema**. Los requisitos pueden variar desde declaraciones abstractas de alto nivel de servicios o restricciones del sistema hasta especificaciones funcionales matemáticas detalladas.

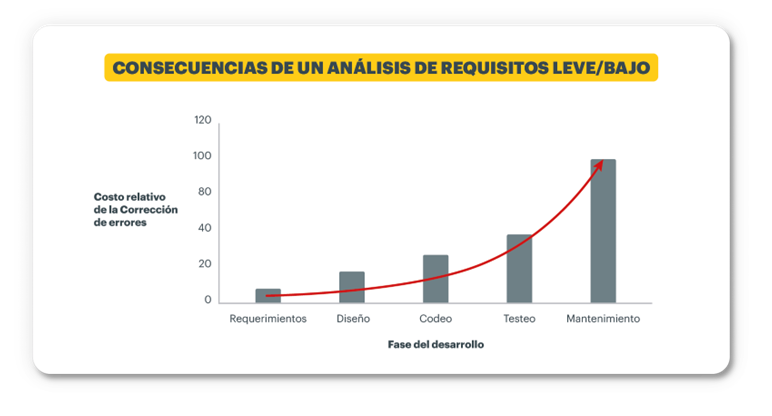
¿Qué es el análisis de requisitos? Es el proceso por el que se determinan las expectativas del usuario para un sistema en consideración. El resultado de este análisis debe ser cuantificable y detallado.

* Sirve como base para los planes de prueba y el plan del proyecto.
* Sirve como un acuerdo entre el desarrollador y el cliente.
* Es un proceso que permite aclarar los requisitos declarados y no declarados
* Sirve para validar el requisito de integridad, falta de ambigüedad y viabilidad.



#### Imagen 7.1: El impacto de un error en el análisis de requerimientos. Fuente: Foundations of software testing de Dorothy Graham, Rex Black, Erik Van Veenendaal e Isabel Evans.

En la imagen 7.2 se muestra la consecuencia de un análisis deficiente de los requisitos y su impacto en el ciclo de vida del desarrollo del software.



#### Imagen 7.2: Consecuencias en el costo de un análisis deficiente. Fuente: elaboración propia.

El cuadro muestra la relación entre la fase del desarrollo y el costo de la corrección de errores. Si el análisis de requisitos no se realiza en la fase inicial del SDLC, entonces su impacto es enorme para solucionarlo en fases posteriores. Cuando el análisis de requisitos no se produce de forma satisfactoria, pueden aparecer algunas consecuencias como ser: la entrega incorrecta de funciones, mala calidad del producto, una gran cantidad de cambios para corregir fallas del sistema, extensión de los plazos iniciales del proyecto, entre otros. Cuanto más se extiende el tiempo en analizar el requisito, más cuesta hacerlo luego y eso afecta la entrega y la calidad del proyecto.

### Desafíos en la fase de análisis de requisitos en el control de calidad

A continuación presentamos algunos de los desafíos más habituales en el momento de poder analizar los requerimientos. En la teoría parece simple lograr este nivel de comunicación entre los equipos, pero esta etapa del proceso suele ser la que más desafíos trae.

* En la etapa inicial de SDLC, el alcance no está claramente definido.
* Muchas veces hay una comprensión ambigua de los procesos.
* La comunicación entre el equipo del proyecto y las partes interesadas juega un papel importante.
* Las entradas insuficientes del cliente conducen a suposiciones y no se aceptan en UAT[[3]](#footnote-2).
* Inconsistencia dentro de un solo proceso en múltiples usuarios
* Puntos de vista conflictivos de los clientes.
* Nuevos requisitos frecuentes.

En los casos vistos hasta el encuentro de hoy, los requerimientos siempre se presentaron como una lista muy ordenada de elementos a desarrollar. La vida profesional está más cerca de esta lista de desafíos y pasa el tiempo entre que el cliente presenta su lista de requerimientos y se llega a una lista más definida con la que se puede trabajar sin cambios.

### Herramientas y técnicas utilizadas para el análisis de los requisitos

Para mitigar el efecto de un mal análisis de requerimientos, existen herramientas y técnicas que podemos aplicar en este momento. Te presentamos dos:

1. **Casos de Uso:** Es una metodología utilizada en el análisis de requisitos para identificar, aclarar y organizar los requisitos. Es un conjunto de posibles secuencias de interacciones entre sistemas y usuarios en un entorno particular y relacionado con un objetivo particular.
2. **Documento de comprensión de los requisitos[[4]](#footnote-3) (RUD):** el documento cubre los detalles de la comprensión de los requisitos relacionados con los puntos siguientes:

* Suposiciones
* Detalles del sistema
* Requisitos del sistema lógico
* Entidad del sistema
* Hardware
* Criterios de aceptación
* Priorizar cada requisito
* Discutir con el equipo e identificar el alcance de la prueba
* Desglose los requisitos en tareas e historias de usuario.

### ¿Cómo analizar los requisitos?

* Averigua qué software tienes que hacer.
* Identifica los requisitos preguntando por qué, qué, quién, cómo, etc.
* Descubre cuán compleja sería la aplicación y su impacto en las pruebas.
* Ten en cuenta que todas las cosas tendrían que ser probadas.

### Validación de requisitos

Valida los requisitos en función de los puntos a continuación para que al final de la fase de análisis de requisitos esté disponible toda la información requerida.

* Corrección: averiguar declaración/requisito incorrecto.
* Completitud: tienes el desafío de encontrar el requisito faltante.
* Factibilidad: encuentra todas las características que son posibles de probar y cuáles están más allá del alcance.
* Testeabilidad: verifica la posibilidad de crear diferentes pruebas aplicables.
* No Ambigüedad: encuentra una interpretación única de los requisitos.
* Consistencia: descubre la consistencia del requisito y apunte a un solo requisito.

**¡MANOS A LA OBRA! **

A continuación encontrarás una serie de requisitos (son los mismos que figuraban como **pregunta bonus** en el Integrador).

**Desafío 7.1:**

1. Relee los requerimientos para poder clasificarlos en 3 tipos:

* Requerimientos funcionales (ej: imprimir un reporte en forma correcta)
* Requerimientos no funcionales (ej: imprimir el reporte en menos de 1 minuto)
* Requerimientos especiales (ej: roles especiales de usuario)

1. Realiza un cuadro o resumen en donde vuelques todos los requerimientos que hayas identificado, con una columna para el número de requerimiento y otra para la descripción.

**Tip:** este ejercicio de categorización te ayudará inmensamente en el proceso de análisis de requerimientos.

**Lista de requerimientos a categorizar:**

* El cliente debe estar ingresado en la empresa como cliente para poder generar el pedido.
* Debe existir un catálogo de productos, indicando diseño, formato, marca, tamaños, colores, y lugares propuestos donde puede imprimirse el logo.
* Dentro del pedido debe existir un botón para ir a definir el diseño del producto, adjuntando el logo del producto en archivos tipo: jpg o png. El sistema no podrá procesar otro tipo de archivos como ser gif, PDF u otros.
* A cada pedido se le asignará un identificador o número único y correlativo, que será utilizado para hacer el seguimiento en todos los procesos siguientes que se realicen.
* El proceso de compras en el sistema abarcará los siguientes pasos y transacciones: Ingreso del pedido, emisión de cotización del pedido para el cliente y revisión del pedido. Allí el cliente podrá eliminar o agregar cantidades. Si desea agregar nuevos productos, deberá generar un nuevo pedido.
* La contabilización de las facturas de venta y facturas de compra podrán configurarse para realizarse de forma automatizada, o podrá ingresarlas manualmente con acceso de un empleado con nivel de supervisor, para cuando se corte la luz en la sucursal o algunas otras excepciones. Estas facturas no serán impresas, solo se podrán guardar como archivo pdf ya que se realizarán manualmente las facturas en el mismo papel impreso. El archivo pdf deberá ser enviado al teléfono del cliente.

**MATERIAL DE LECTURA **

### Función de control de calidad

Un miembro de un equipo QA debe estar involucrado en la actividad de análisis de requisitos para garantizar que los requisitos identificados por el cliente o el BA[[5]](#footnote-4) y aceptados por el cliente sean medibles. Además, esta actividad proporciona claridad en varias etapas de SDLC ya que permite identificar la disponibilidad de recursos, permite anticipar los tiempos y preparación de pruebas.

A continuación, las actividades que debe realizar el control de calidad:

1. Analiza todos y cada uno de los requisitos del documento de especificación. Cuando puedas y tengas suficiente información, redacta los casos de uso.
2. Enumera escenarios de alto nivel.
3. Aclara tus consultas y la funcionalidad con las partes interesadas.
4. Promueve sugerencias para implementar las funciones o cualquier problema lógico.
5. Identifica defectos o necesidades de aclaración contra el pliego de condiciones presentado por el cliente.
6. Realiza el seguimiento del defecto (si los hubiera detectado) con respecto al documento de especificaciones.
7. Crea escenarios de prueba de alto nivel.
8. Crea una Matriz de Trazabilidad.

Para que un equipo de QA pueda iniciar este proceso de análisis de requerimientos debe contar con un documento, habitualmente llamado SRS (Software Requirement Specification).

En esta fase, el equipo de control de calidad (QA) analiza a un nivel superior qué probar y cómo probar. Dicho de otro modo, diseñan qué pruebas se deben realizar, en qué momento y con qué herramientas.

El equipo de control de calidad luego realiza un seguimiento con varias partes interesadas, como Business Analyst, System Architecture, Client, Test Manager/Lead, en caso de que se requiera alguna consulta o aclaración para comprender el requisito. Los requisitos pueden ser funcionales o no funcionales, como rendimiento, seguridad, facilidad de uso, etc., o pueden ser ambas cosas al mismo tiempo.

¿Para qué se realiza este seguimiento? El equipo de QA está buscando completar una RTM o Matriz de trazabilidad. Además buscará realizar el informe de factibilidad de automatización y una lista de preguntas, si corresponde, para ser más específicos sobre los requisitos.

### Actividades realizadas para el análisis de requisitos

Hay tres actividades principales que realiza el equipo de control de calidad en esta fase. Las actividades se describen a continuación.

**DEFINICIÓN DEL ALCANCE**

El equipo de control de calidad identifica el alcance de las pruebas en niveles altos y se divide en varios módulos funcionales. El equipo también identifica los tipos de pruebas necesarias para realizar: pruebas de humo[[6]](#footnote-5), pruebas de cordura[[7]](#footnote-6), pruebas funcionales, pruebas de regresión, etc.

El equipo de control de calidad analiza los requisitos previos y los detalles del entorno donde se supone que se realizarán las pruebas. El equipo recopila detalles sobre las prioridades de las pruebas y se enfoca en la secuencia de módulos que se validarán.

**RTM (MATRIZ DE TRAZABILIDAD)**

El seguimiento de requisitos es un proceso de documentación de los vínculos entre los requisitos y los productos de trabajo desarrollados para implementar y verificar esos requisitos. El RTM captura todos los requisitos en el Análisis de Requisitos junto con su trazabilidad en un solo documento. Todo esto se entrega al final del ciclo de vida.

La Matriz se crea al comienzo de un proyecto, ya que forma la base del alcance del proyecto y los entregables que se producirán. Es bidireccional, ya que realiza un seguimiento del requisito hacia adelante al examinar la salida de los entregables y hacia atrás al observar el requisito comercial que se especificó para una característica particular del producto.

**¿QUÉ ES LA MATRIZ DE TRAZABILIDAD DE REQUERIMIENTOS (RTM)?**

La Matriz de trazabilidad de requerimientos es un documento que mapea y rastrea los requerimientos del usuario con casos de prueba. Captura todos los requerimientos propuestos por el cliente y la trazabilidad de los requerimientos en un solo documento, entregado al finalizar el ciclo de vida del desarrollo del Software, ya que es un entregable.

El objetivo principal de la Matriz de trazabilidad de requerimientos es validar que todos los requerimientos se verifiquen a través de casos de prueba, de modo que no se desmarque ninguna funcionalidad durante las pruebas de software.

Surge aquí una pregunta, teniendo en cuenta todos los escenarios / casos posibles. ¿Cómo asegurarse de que ningún requisito se quede fuera del ciclo de prueba? Usaremos las siguientes herramientas.

Un **escenario de prueba** es la representación de uno o más flujos de negocio o flujos críticos que se ejecutarán de forma concurrente sobre la infraestructura objetivo por uno o más usuarios virtuales.

Un **caso de prueba** es un documento/plantilla cargada en un sistema de gestión de pruebas, que consiste en un conjunto de variables y condiciones en las cuales la viabilidad de una aplicación de software debe predeterminarse para verificar su funcionalidad. Ayuda a los testers a determinar si un sistema está funcionando como debía funcionar según los requisitos del cliente. Si el caso de prueba es el "cómo", entonces el escenario de prueba es el "qué". Es una secuencia de muchos casos de prueba que deben ejecutarse uno tras otro para verificar la funcionalidad de la aplicación. Los casos de prueba son un subconjunto de escenarios de prueba, mientras que el último se basa en la funcionalidad y se trata del flujo de trabajo.

| En los próximos encuentros estaremos aprendiendo sobre **casos de prueba**. Mientras tanto te dejamos esta definición para darte más contexto: Un caso de prueba es el diseño de cómo y bajo qué condiciones se probará algo. Por ejemplo, si deseas probar un automóvil, un **escenario de prueba** es “probar el vehículo en condiciones de tormenta de granizo” y un **caso de prueba** puede ser “Probar el sistema de frenos bajo condiciones meteorológicas de tormenta.” |
| --- |

Ejemplos de escenarios de prueba es un sistema de ventas al exterior o multinacional:

* Validar si el administrador del sistema puede agregar un nuevo país
* Validar si el administrador del sistema puede eliminar un país existente
* Validar si se puede actualizar un país existente mal escrito
* Validar que se pueda vender a cualquier país con el que tengamos convenios/contratos de exportación
* Validar que no se pueda vender a países donde no hay convenios o contratos para exportar

**¿Por qué es importante la RTM?**

Es una forma sencilla de rastrear el requisito con sus escenarios de prueba y casos de prueba correspondientes. RTM suele ser una hoja de trabajo que contiene los requerimientos con todos los escenarios y casos de prueba posibles y su estado actual, es decir, si se aprobaron o fallaron. Esto ayudaría al equipo de prueba a comprender el nivel de actividades de prueba realizadas para el producto específico.

**¿Qué parámetros debemos incluir en la RTM?**

ID de requerimiento (ID: es la identificación o número/letras que lo identifica al requisito, el cuál es único, no se repite)

Tipo de requisito y descripción

Casos de prueba con estado

Ejemplo: un ejemplo simple de Matriz de trazabilidad de requerimientos, donde podemos indicar el resultado de la prueba en el Status, si Pasó o Pass: es que no hay errores, si falló o Fail: es que hay errores, y No run: no pudo ser corrida, el sistema no lo permitió o no existe la opción para hacerlo.



#### Imagen 7.3: Ejemplo RTM. Fuente: elaboración propia.

**¿Qué vemos en la matriz de trazabilidad?**

Una matriz de trazabilidad de requerimientos permite mostrar:

* Cobertura de requerimientos con el número de casos de prueba, normalmente se expresa en %
* Estado de diseño y estado de ejecución para un caso de prueba específico
* Los defectos relacionados y el estado actual también se pueden mencionar en la misma matriz.
* Este tipo de matriz proporcionaría una pantalla para todas las actividades de prueba.
* Un equipo de testers también puede optar por el seguimiento de requerimientos en las herramientas de gestión de pruebas disponibles.
* Confirma la cobertura de prueba del 100%, es decir que se han corrido un 100% de las pruebas diseñadas.
* Indica los requerimientos que faltan probar o las incoherencias del documento.
* Muestra los defectos generales o el estado de ejecución con un enfoque en los requerimientos del negocio.
* Ayuda a analizar o estimar el impacto en el trabajo del equipo de control de calidad con respecto a la revisión o reelaboración de los casos de prueba (volver a escribirlos revisando bien los requerimientos y hablando con el desarrollador para acordar los objetivos).

**¿Cómo crear una matriz de trazabilidad de requerimientos?**

Ya hemos realizado en el ejercicio 7.1 el primer paso: documentar todos los requerimientos identificados en el caso y asignarles un número de seguimiento.

**¿NECESITAS UN EJEMPLO? **

Tomaremos un proyecto ficticio: Bank007.

Sobre la base del Documento de requerimientos del negocio (BRD - *business requirements document*) y el Documento de requerimientos técnicos (TRD - *technical requirements document*), los testers comienzan a escribir casos de prueba.

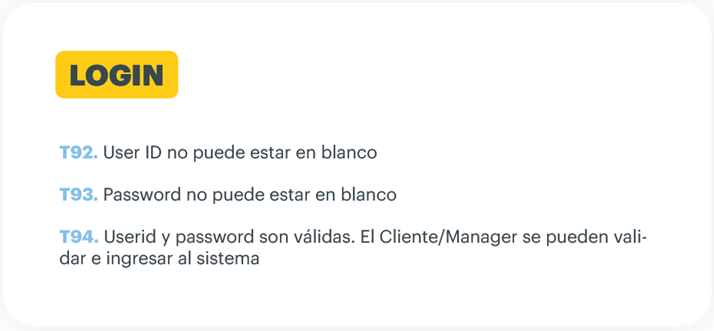
La siguiente tabla es nuestro documento de requerimientos del negocio o BRD para el proyecto bancario Bank007.

El cliente debería poder iniciar sesión en el sitio web bancario Bank007 con la contraseña y el número de identificación de usuario correctos. El administrador también debería poder iniciar sesión en el sitio web a través de un inicio de sesión idéntico que le de acceso a más funcionalidades.



#### Imagen 7.4: Ejemplo casos de prueba. Fuente: elaboración propia.

La siguiente tabla es nuestro Documento de requerimientos técnicos (TRD).



#### Imagen 7.5: Documento de requerimientos técnicos. Fuente: elaboración propia.

| Nota: Los equipos de Testing no documentan el BRD ni el TRD, lo realizan los Analistas de negocios o de sistemas. Además, algunas empresas utilizan Documentos de requerimientos de función (FRD - *functional requirements document*) que son similares al Documento de requerimientos técnicos, pero el proceso de creación de la Matriz de trazabilidad sigue siendo el mismo. |
| --- |

**Ahora tenemos toda la información necesaria para poder crear una RTM en pruebas:**

* Paso 1: Nuestro caso de prueba de muestra es "Verificar el inicio de sesión, cuando se ingrese el ID y la contraseña correctos, debe iniciar sesión correctamente"



#### Imagen 7.6: RTM - Paso 1. Fuente: elaboración propia.

* Paso 2: Identifique el requisito técnico que este caso de prueba está verificando. Para nuestro caso de prueba, se está verificando el requisito técnico es T94. T94 Si el User id y el password son válidos, login exitoso
* Paso 3: tengamos en cuenta este requisito técnico (T94) en el caso de prueba.



#### Imagen 7.7: RTM - Paso 3. Fuente: elaboración propia.

* Paso 4: Identificar el requisito del negocio para el que se define este TR (requisito técnico-T94)



#### Imagen 7.8: RTM - Paso 4. Fuente: elaboración propia.

* Paso 5: tenga en cuenta el BR (requisito del negocio) en el caso de prueba



#### Imagen 7.9: RTM - Paso 5. Fuente: elaboración propia.

Identificar el requerimiento para el cual el requisito técnico T94 fue definido

* Paso 6: haga lo anterior para todos los casos de prueba. Más tarde, extraiga las primeras 3 columnas de su conjunto de pruebas. ¡RTM en prueba está listo!



#### Imagen 7.10: RTM en prueba - Paso 6. Fuente: elaboración propia.

**¡MANOS A LA OBRA! **

Si estás con tiempo, ¿te animas a continuar el análisis de requisitos que realizaste en el primer ejercicio y realizar un prototipo de BRD? Un tester no suele redactar este documento pero existe la posibilidad de que tengas experiencia en negocios y sea de tu interés capacitarte en la creación de este documento. En especial si deseas tener un desempeño profesional mixto[[8]](#footnote-7).

**MATERIAL DE LECTURA **

**ANÁLISIS DE AUTOMATIZACIÓN**

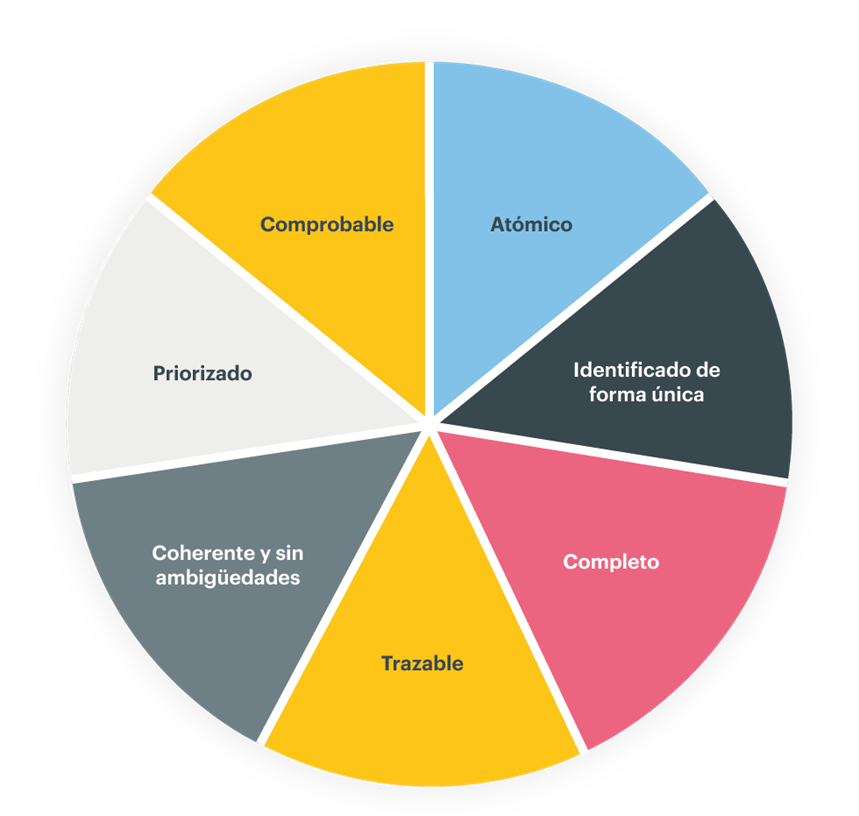
La automatización es una optimización que se realiza para poder replicar las pruebas una y otra vez cada vez que una nueva versión de software está disponible para ser testeada. Para automatizar se requieren conocimientos de programación ya que los QA especializados en automatización escriben sus propias pruebas para cada caso que deseen testear.

En la fase de requisitos, el equipo de control de calidad analiza el alcance de la automatización para las **pruebas de regresión**[[9]](#footnote-8). Si se agrega la automatización al alcance, el equipo decide qué herramienta se puede usar, qué funcionalidades se cubrirán como automatización, el marco de tiempo y la asignación de recursos involucrados para el desarrollo de la automatización. Una vez que se completa este análisis, el equipo de control de calidad proporciona el Informe de viabilidad de automatización a diferentes partes interesadas para que lo aprueben.

**Cómo analizar los requisitos**

Un requisito bien desarrollado debe mantener estas características. Debe ser:

* Atómico
* Identificado de forma única
* Completo
* Coherente y sin ambigüedades
* Trazable
* Priorizado
* Comprobable



#### Imagen 7.11: Características de los requisitos. Fuente: elaboración propia.

**¿NECESITAS UN EJEMPLO? **

Considera un ejemplo de un sistema de software educativo donde un estudiante puede registrarse para diferentes cursos.

A continuación te presentamos una tabla con tres columnas:

1. La primera columna indica la calidad del requisito.
2. La segunda columna indica el requisito mal formulado.
3. La tercera columna es igual a la segunda columna, pero el requisito fue transformado a un buen requisito.



#### Imagen 7.12: Ejemplos de requisitos. Fuente: elaboración propia.

**Ahora podemos inferir el significado de cada uno de estos atributos de un buen requisito:**

#### Atómico

Todos y cada uno de los requisitos deben ser atómicos. Es decir, deben tener un nivel de detalles muy bajo y no debería ser posible separarlos en componentes.

Continuando con el ejemplo anterior, el mal requisito es “Los estudiantes podrán matricularse en cursos de grado y posgrado”. Este es un mal requisito porque no es atómico ya que habla de dos entidades[[10]](#footnote-9) (diferentes, cursos de pre grado y posgrado. La forma en la que este requisito se vuelve atómico es al separarlo en dos: un requisito para cada entidad. Uno hablará de la inscripción a los cursos de pre grado, mientras que el otro hablará de la inscripción a los cursos de posgrado.

#### Identificado de forma única

De manera similar, el siguiente atributo de calidad es verificar la identificación única. En el ejemplo que venimos usando, tenemos dos requisitos separados, pero ambos tienen el mismo ID # 1. Por lo tanto, al hacer referencia al ID#1, no sabremos con exactitud a qué porción del requerimiento estamos haciendo referencia. Entonces, separarlos con identificaciones únicas nos dará como resultado lo siguiente:

Sección 1: inscripciones en cursos.

Requisito: 1.1 es inscripción en cursos de pre grado.

Requisito: 1.2 es inscripción en cursos de posgrado.

#### Completo

Además, todos y cada uno de los requisitos deben estar completos. En la imagen 7.12, vemos cómo el ejemplo de un requisito incompleto dice que un "usuario profesor iniciará sesión en el sistema proporcionando su nombre de usuario, contraseña y *otra información relevante*". Aquí, la “otra información relevante” no está clara, por lo que la otra información relevante debe detallarse para completar el requisito.

#### Consistente y sin ambigüedades

Los requisitos deben ser consistentes e inequívocos. En el cuadro vemos que coexisten dos requisitos que presentan contradicciones.

* "Un estudiante tendrá cursos de pre grado o cursos de posgrado, pero no ambos".
* "Algunos cursos tendrán que estar abierto tanto a estudiantes de pre grado como de posgrado”.

El problema en este requisito es que desde el primer requisito parece que los cursos se dividen en dos categorías: cursos de pregrado y cursos de posgrado y el estudiante puede optar por cualquiera de los dos, pero no por ambos. Pero cuando lee otro requisito, entra en conflicto con el primer requisito y dice que algunos cursos se abrirán tanto para graduados como para estudiantes pre-grado.

Por lo tanto, es necesario convertir este requisito en un buen requisito, que es "Un estudiante tendrá cursos de pre grado o cursos de posgrado, pero no ambos[[11]](#footnote-10)". Lo que significa que cada curso se marcará como curso de pre grado o curso de posgrado.

#### Trazable

Todos y cada uno de los requisitos deben ser rastreables porque existen diferentes niveles de requisitos. Más adelante veremos que estos niveles son: requisitos comerciales, requisitos arquitectónicos y de diseño; seguidos de requisitos de integración del sistema.

Ahora, cuando convertimos los requisitos comerciales en requisitos arquitectónicos y de diseño o convertimos los requisitos arquitectónicos y de diseño en requisitos de integración de sistemas, tiene que haber trazabilidad. Lo que significa que deberíamos ser capaces de tomar todos y cada uno de los requisitos comerciales y asignarlos a uno o más requisitos de arquitectura y diseño de software correspondientes. Entonces, aquí hay un ejemplo de un requisito incompleto que dice: "¿Mantener la información del estudiante, asignada a la ID de solicitud de BRD?" Asignando la identificación del requisito 4.1.ya tenemos un requisito trazable.

Por lo tanto, este mapeo debe estar ahí para todos y cada uno de los requisitos. De la misma manera que tenemos un requisito de mapeo de alto y bajo nivel, el mapeo también existe entre el sistema y el requisito de integración con el código que implementa ese requisito y también hay un mapeo entre el sistema y el requisito de integración con el caso de prueba que prueba ese requisito en particular. Esto permite la trazabilidad a lo largo de todo el proyecto

#### Priorizado

Luego, se deben priorizar todos y cada uno de los requisitos, de modo que el equipo tenga una guía sobre qué requisito se puede implementar primero y cuál se puede hacer más adelante. En el ejemplo vemos el caso común de que todos los requisitos son importantes y han sido priorizados con la misma prioridad 1.

Todo no puede tener la misma prioridad, ya que los equipos deben poder saber por dónde empezar o cuál es el requisito del cual dependen otros.

Entonces, el ejemplo de un buen requisito aquí es el registro de estudiantes y la inscripción de cursos tiene la prioridad más alta 1, mientras que mantener la información del usuario está debajo en la prioridad 2 y luego tenemos ver la boleta de calificaciones en la prioridad 3.

#### Comprobable

Todos y cada uno de los requisitos deben ser comprobables.

En nuestro ejemplo, el requisito mal redactado es "cada página del sistema se cargará en un marco de tiempo aceptable". Ahora, hay dos problemas con este requisito: primero, cada página significa que puede haber muchas páginas, lo que hará que los esfuerzos de prueba sean difíciles de medir. El otro problema es que dice que la página se va a cargar en un período de tiempo aceptable. Ahora, ¿cuál es el período de tiempo aceptable? ¿Aceptable para quién? Entonces, tenemos que convertir un argumento no comprobable en un argumento comprobable, que indique específicamente de qué página estamos hablando ("páginas de registro de estudiantes e inscripción de cursos") y también se proporciona el marco de tiempo aceptable (5 segundos).

**Conclusión**

Esta es una mera introducción a la buena práctica en la redacción de requisitos. El hábito de ir redactando requisitos atómicos, completos e identificables acelera la comunicación entre los equipos y ahorra tiempo y desgaste.

**MATERIAL DE LECTURA **

**Resultado de la fase de análisis de requerimientos**

Luego de completar la fase de “Análisis de requerimientos”, tendremos esta documentación[[12]](#footnote-11) a disposición de los equipos:

* Documento de comprensión de requisitos.
* Escenarios de alto nivel.
* Estrategia de prueba de alto nivel y aplicabilidad de prueba.

**Análisis de requisitos de software aplicado**

En el encuentro de hoy dedicamos mucho tiempo a los requerimientos. ¿La razón? Un buen análisis de requerimientos ahorra tiempo, elimina errores, baja la cantidad de suposiciones entre los equipos y entrega felicidad a los desarrolladores[[13]](#footnote-12).

El requisito de software detalla las necesidades funcionales o no funcionales que deben implementarse en un sistema.

**Funcional** significa proporcionar un servicio particular al usuario.

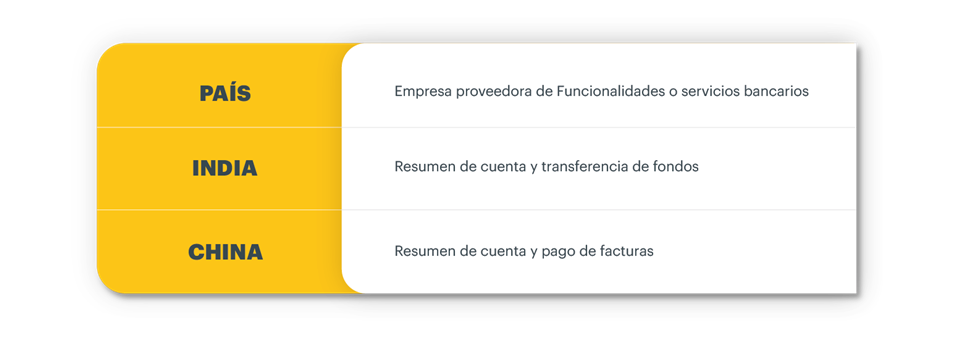
Por ejemplo, en el contexto de la aplicación bancaria, el **requisito funcional** será que cuando el cliente seleccione "Ver saldo", debe poder ver el último saldo de su cuenta.

El requisito de software también puede ser **no funcional**. Puede ser un requisito de rendimiento. Por ejemplo, un requisito no funcional es que cada pantalla de aviso del sistema sea visible para los usuarios por 5 segundos[[14]](#footnote-13).

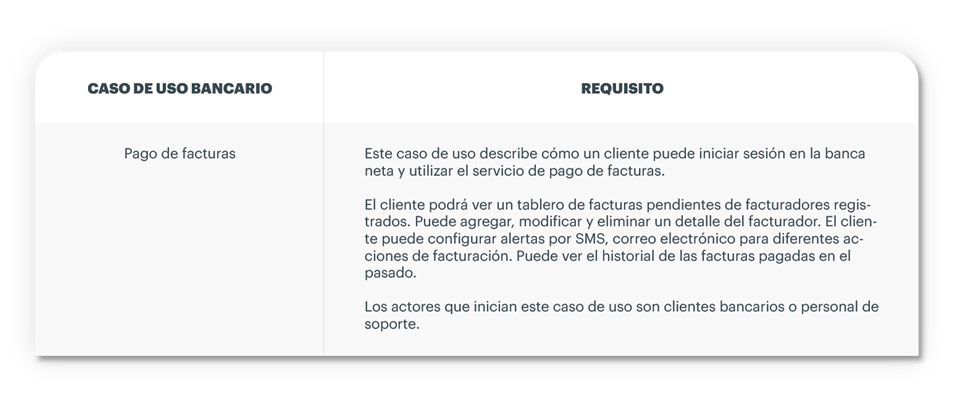
**Para claridad y especificidad, los requisitos de software generalmente se expresan como declaraciones.**

**Tipos de requisitos**

Requisitos comerciales: son requisitos de alto nivel que se toman del caso comercial de los proyectos. Por ejemplo, un sistema de servicios bancarios móviles brinda servicios bancarios en el sudeste asiático. El requisito comercial que se decide para India es el resumen de cuenta y la transferencia de fondos, mientras que para China es el resumen de cuenta y el pago de facturas.

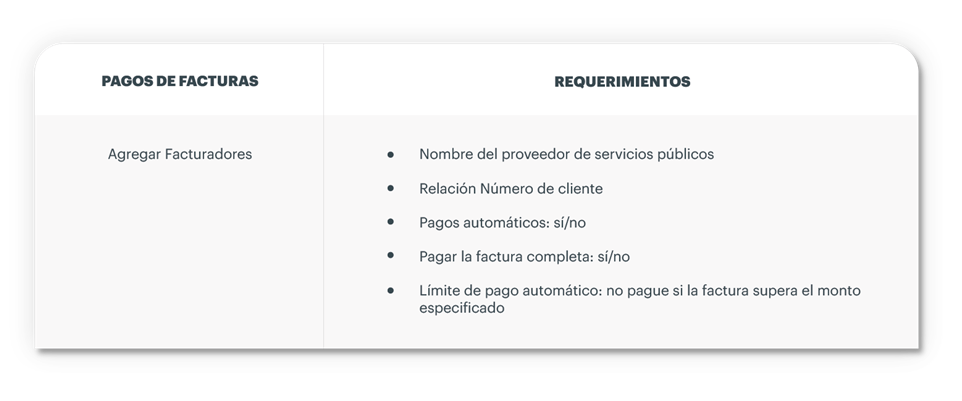


#### Imagen 7.13: Ejemplos de tipos de requisitos. Fuente: elaboración propia.

Requisitos arquitectónicos y de diseño: estos requisitos son más detallados que los requisitos comerciales. Determina el diseño general requerido para implementar el requisito comercial. Para nuestra organización educativa, los casos de uso de arquitectura y diseño serían inicio de sesión, detalles del curso, etc. El requisito sería como se muestra a continuación. 

#### Imagen 7.14: Caso de uso bancario y requisito. Fuente: elaboración propia.

Requisitos del sistema y de integración: En el nivel más bajo, tenemos requisitos del sistema y de integración. Es una descripción detallada de todos y cada uno de los requisitos. Puede ser en forma de historias de usuarios que realmente describen el lenguaje comercial cotidiano. Los requisitos se encuentran en abundantes detalles para que los desarrolladores puedan comenzar a codificar. Aquí, en el ejemplo del módulo de pago de facturas, donde se mencionarán los requisitos para agregar un emisor de facturas.



#### Imagen 7.15: Pago de facturas y requerimientos. Fuente: elaboración propia.

A veces, para algún proyecto, es posible que no recibas ningún requisito o documento para trabajar. Pero todavía hay otras fuentes de requisitos que puedes considerar para el requisito o la información, de modo que se pueda basar el software o diseño de prueba en estos requisitos.

Entonces, las otras fuentes de requisitos en las que puede confiar son

* Transferencia de conocimientos de colegas o empleados que ya trabajan en ese proyecto
* Habla sobre el proyecto con el analista comercial, el gerente de producto, el líder del proyecto y los desarrolladores.
* Analiza la versión anterior del sistema que ya está implementada en el sistema
* Analiza el documento de requisitos anterior del proyecto.
* Mira los informes de errores anteriores, algunos de los informes de errores se convierten en solicitudes de mejora que pueden implementarse en la versión actual.
* Mira la guía de instalación si está disponible para ver cuáles son los requisitos de instalación.
* Analiza el dominio o el conocimiento de la industria que el equipo está tratando de implementar

Independientemente de la fuente de requisitos que obtengas, asegúrate de documentarlos[[15]](#footnote-14) de alguna forma. Haz que otros miembros del equipo con experiencia y conocimientos los revisen.

**¡MANOS A LA OBRA! **

# 7.3 ESCENARIOS DE ANÁLISIS

Te presentamos varios escenarios de análisis en donde veremos aplicados algunos de los conceptos que estudiamos hoy. Te invitamos a que tomes algunos minutos para evaluar el desafío con tus compañeros de equipo y discutan cada escenario y su respuesta. Cada uno deberá exponer las razones por las que elige cada opción. Podrán coincidir o no. Lo importante no es tener resultados unánimes sino entender las razones detrás de cada punto de vista.

Hacia el final de [este formulario](https://docs.google.com/forms/d/e/1FAIpQLScYnYEhVuBOZnrm09VvZQRK9ZxMTj0uUpBlUY7XbAL7V1gXJA/viewform?usp=sf_link) encontrarás estos mismos escenarios y sus respuestas. Aseguráte de enviarlo en forma individual luego de haberlo resuelto. En ese mismo formulario, también encontrarás otras preguntas relacionadas a los temas vistos durante el día de hoy. ¡Complétalo para saber cómo has avanzado en tu aprendizaje! Como siempre, puedes tener en cuenta a tu equipo para conversar sobre las soluciones a las que has llegado.

1. Juan debe realizar un proceso de prueba. Considera que dentro de los pasos a realizar hará la planificación, luego la ejecución y por último el control y monitoreo. Esto es:
   1. Correcto, ya que el orden de las pruebas es inherente al resultado.
   2. Incorrecto, ya que los pasos de prueba deben realizarse en orden.
   3. Incorrecto, ya que la única prueba que debe hacer es el análisis de requerimientos.
   4. Correcto, ya que las pruebas necesitas de estos 3 pasos solamente.
2. Joaquín, que conforma un equipo de control de calidad, debe realizar un análisis de requisitos. Dentro de las actividades que realizará, piensa utilizar una Matriz de Trazabilidad (RTM). Esto es:
   1. Recomendable, ya que esto le permitirá mapear y rastrear los requerimientos del usuario con casos de prueba.
   2. No recomendable, ya que la RTM se utiliza sólo para casos de gran complejidad y se desaconseja su uso para otras instancias.
   3. Recomendable, ya que al utilizar la RTM no deberá realizar los pasos siguientes.
   4. Obsoleto, ya que la RTM sólo funciona para analizar necesidades de los productos.
3. Candela es QA Senior y está liderando un proyecto de control de calidad de una App móvil. Decide mantener informada a las partes y a su equipo de los cambios y avances en el proyecto. Esto es:
   1. No recomendable, ya que Candela tiene más experiencia y no debería dar aviso a nadie.
   2. Correcto, ya que la comunicación con las partes interesadas juega un rol transcendental.
   3. Incorrecto, ya que Candela sólo debe brindar información a su equipo de trabajo.
   4. Recomendable, ya que la comunicación entre el equipo del proyecto y las partes interesadas juega un papel importante.
4. Julieta tiene que analizar los requisitos de Software para una Web de empleos activa. Se le indica que analice los requisitos anteriores a la implementación del proyecto ya que no cuentan con una actualización de los mismos. Esta situación:
   1. Es inadmisible, ya que los requisitos tienen que estar actualizados para que Julieta pueda proceder con el análisis.
   2. No es la ideal, pero Julieta de todas formas podría analizar los requisitos anteriores y documentar los que vayan surgiendo de ese análisis.
   3. No es posible, ya que el análisis sólo se puede hacer en base a los requisitos formales brindados por la empresa.
   4. Es indiferente ya que en esta situación Julieta debería desestimar esos requerimientos y delegar la tarea.
5. Esteban pertenece a una multinacional donde le asignaron llevar el control de calidad de una Web de ventas de zapatos no muy conocida. Dentro de las actividades a realizar, creará escenarios de prueba de bajo nivel. Esto es:
   1. Correcto, ya que los escenarios de alto nivel sólo se utilizan para marcas o empresas importantes.
   2. Incorrecto, ya que debe realizar escenarios de alto nivel y luego verificar cuales son las pruebas necesarias para implementar.
   3. Admisible, ya que ahorra tiempo y presupuesto.
   4. Indiferente, porque los escenarios de prueba no son relevantes en el control de calidad.

# EJERCICIOS DE CONSOLIDACIÓN

Vamos a integrar todo lo que hemos visto en el encuentro de hoy con los siguientes ejercicios.

### Tarea #1

Analizaremos la descripción del pedido del cliente y haremos una lista de los requerimientos o requisitos que podamos descubrir en el texto.

Ejemplo:

Requerimiento: El sistema deberá permitir registrar usuarios

### Tarea #2

Elegiremos una herramienta para crear diagramas de casos de uso.

### Tarea #3

Identificaremos los posibles roles involucrados en todo el proceso y crearemos un diagrama de casos de uso con los resultados.

### Tarea #4

Arma un cuadro con los requisitos e intenta redactarlos siguiendo las buenas prácticas. Nota: ten paciencia. Esta habilidad requiere **mucha** práctica.

**Descripción del pedido del cliente**

Los clientes acceden a la información sobre los libros a través de la Web y realizan búsquedas por autor, título o editorial. A medida que navegan por las distintas páginas, y encuentran algún libro que les interesa, lo incluyen en el carrito de la compra para efectuar al final el pedido correspondiente. Para realizar un pedido, un cliente debe estar previamente registrado como tal. Esto significa introducir una serie de datos personales (nombre y apellidos, dirección, localidad, código postal, país), datos de la tarjeta de crédito (tipo de tarjeta, número, fecha límite de validez) y sobre preferencias de envío (correo normal, expreso, internacional). Asociado a un pedido específico pueden introducirse opciones de empaquetado (estándar o regalo), tarjeta con mensaje adicional cuando es un regalo, o un nombre y dirección de otra persona a la que se le hace enviar un pedido. Como es habitual en este tipo de aplicaciones, debería elegir un nombre de usuario y una clave como método de autentificación para efectuar las transacciones habituales con la librería. Cuando se han incluido en el carrito de la compra el conjunto de los libros deseados (cantidad, título y autor), se debe pasar al proceso de confirmar el pedido que debería requerir un paso previo de seguridad para garantizar que el cliente es quien dice ser. Una vez introducidos todos los datos adicionales, el cliente confirma el pedido que pasa a un estado de espera -90 minutos- durante el cual es posible modificar algunos de los ítem del pedido (eliminar o cambiar cantidad) pero no añadir nuevos ítem, para lo cual se debería crear un nuevo pedido. Una vez transcurridos los 90 minutos, los pedidos quedan confirmados definitivamente y no se pueden modificar ni anular. La empresa puede realizar envíos parciales en función de la disponibilidad de los ítem, pero sin modificar el costo total de envío debido a este fraccionamiento del pedido. A medida que se van armando los pedidos se envía un e-mail al cliente para confirmar el pedido, lo mismo que al realizar el envío correspondiente.

1. Puede ser que estés en un contexto de desarrollo ágil de software pero también existen contextos más tradicionales, o *waterfall* que siguen procesos más delimitados. [↑](#footnote-ref-0)
2. ¿Tienes espacio para ver un video de 8 minutos? [Haz clic aquí](https://www.youtube.com/watch?v=suLnoy9-ueA) para ver cuál puede ser el desempeño profesional de un QA funcional junior. [↑](#footnote-ref-1)
3. UAT o *user acceptance test.* [↑](#footnote-ref-2)
4. Requisitos y requerimientos se usarán indistintamente. Verás que en la industria se habla de requerimientos, features o requisitos según la región o el idioma en el que hablen los equipos. [↑](#footnote-ref-3)
5. BA: Business analyst. En ocasiones es un BA quien trae una propuesta de requerimientos a un equipo de desarrollo. Un BA es parte del equipo de Producto y se orienta a descubrir oportunidades comerciales. Ej: agregar la feature de “Comunidad” a una app para bajar de peso. [↑](#footnote-ref-4)
6. Las traducciones a veces nos dan alegrías. Pruebas de cordura y de humo. [¿Quieres saber más?](https://spa.myservername.com/smoke-testing-vs-sanity-testing) [↑](#footnote-ref-5)
7. A pesar de su nombre, no incluye preguntarle a los integrantes de un equipo cómo se sienten. [Aquí puedes leer más](https://es.education-wiki.com/7357255-sanity-testing) sobre las pruebas de cordura. [↑](#footnote-ref-6)
8. En el mundo profesional de la tecnología los perfiles mixtos son aquellos profesionales que ya tienen un recorrido profesional en alguna área (por ej, han trabajado como administrativos en un banco) y al capacitarse en el mundo digital *suman habilidades* difíciles de encontrar. Saben de negocios, saben de administración y saben de tecnología. Tal vez tú eres uno de ellos. [↑](#footnote-ref-7)
9. Es como tu primera semana en un trabajo nuevo. Todos usan palabras que desconoces. ¡No te preocupes! Aquí estamos para salvar cada duda que puedas tener. [¿Qué son las pruebas de regresión?](https://geekflare.com/es/regression-testing-tools/) [↑](#footnote-ref-8)
10. En software, una entidad es un objeto exclusivo único en el mundo real que se está controlando. Algunos ejemplos de entidad son una sola persona, un solo producto o una sola organización. En este ejemplo: el curso ofrecido. [↑](#footnote-ref-9)
11. Esta es una solución posible. El tester podrá identificar las contradicciones y levantar la alerta. No es responsabilidad del tester resolver cómo se estipulan finalmente los requisitos. [↑](#footnote-ref-10)
12. Documentar es la segunda actividad más importante para un tester. ¿La primera? **Atención al deatlle…** (¿Lo notaste? Lo hicimos para ver si estabas prestando atención) [↑](#footnote-ref-11)
13. Si no estamos para dar felicidad a los desarrolladores y con eso generar productos tecnológicos que valgan la pena y resuelvan problemas… ¿para qué estamos? [↑](#footnote-ref-12)
14. ¿No odias cuando el sistema te informa un mensaje y luego desaparece sin que puedas leerlo? Ahora puedes ser parte de la solución a estos malestares. [↑](#footnote-ref-13)
15. No puedes decir que no te avisamos: documentar, documentar, documentar. [↑](#footnote-ref-14)