Clase 08. EJECUCIÓN DE PRUEBAS

## Proceso prueba de contexto

No existe un único y universal proceso de prueba de software, pero existen conjuntos de actividades de prueba comunes sin las cuales es menos probable que la prueba llegue a los objetivos establecidos. Estos conjuntos de actividades de prueba son un proceso de prueba.

## Actividades y tareas de prueba

1. Planificación de la prueba.
2. Monitorización y control de la prueba.
3. Análisis de la prueba.
4. Diseño de la prueba.
5. Implementación de la prueba.
6. Ejecución de la prueba.
7. Compleción de la prueba.

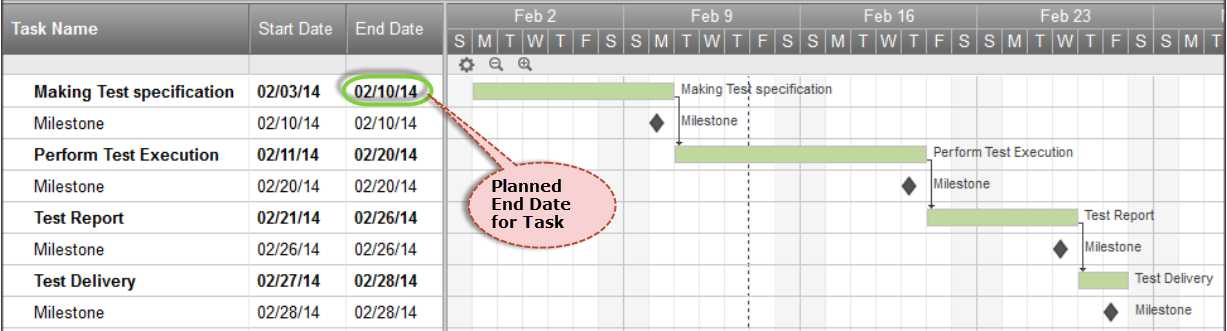
### Planificación de la prueba

La planificación de la prueba implica ciertas tareas que definen sus objetivos y el enfoque para cumplir con los objetivos de la prueba dentro de las restricciones impuestas por el contexto. Los planes de prueba pueden ser revisados en función de la retroalimentación de las actividades de monitorización y control.

#### 

#### Ejemplo:

Calendario de Pruebas



### Fuente: [Guru99](https://www.guru99.com/how-you-can-achieve-project-goals-through-test-monitoring-control.html) **Monitorización y control de la prueba**

La monitorización (o seguimiento) de la prueba implica la comparación continua del avance real con respecto al plan de prueba utilizando cualquier métrica de monitorización de la prueba definida en el plan de prueba. **El control de la prueba implica tomar las medidas necesarias para cumplir los objetivos del plan**

**de prueba.**

El avance de la prueba con respecto al plan se comunica a los implicados por medio de informes de avance de la prueba, incluyendo las desviaciones con respecto al plan y la información que permita apoyar cualquier decisión de interrumpir la prueba.

### Análisis de la prueba

Durante el análisis de la prueba, se analiza la base de prueba para identificar las prestaciones que presentan capacidad de ser probadas y definir las condiciones de prueba asociadas. En otras palabras, el análisis de la prueba determina "qué probar" en términos de criterios de cobertura medibles.

#### Actividades de análisis

* Analizar la base de prueba correspondiente al nivel de prueba considerado.
* Evaluar la base de prueba y los elementos de prueba para identificar defectos de distintos tipos.
* Identificar las prestaciones y conjuntos de prestaciones que se van a probar.
* Definir y priorizar las condiciones de prueba para cada prestación basándose en el análisis completo del sistema y sus pruebas.
* Captura de la trazabilidad bidireccional entre cada elemento de la base de prueba y las condiciones de prueba asociadas.

### Diseño de la prueba

Durante el diseño de la prueba, las condiciones de prueba se transforman en casos de prueba de alto nivel, conjuntos de casos de prueba de alto nivel y otros productos de prueba. Es decir, el análisis de la prueba responde a la pregunta "¿qué probar?" y el diseño de la prueba responde a la pregunta "¿cómo probar?".

#### El diseño incluye

* Diseñar y priorizar casos de prueba y conjuntos de casos de prueba.
* Identificar los datos de prueba para apoyar las condiciones y los casos de prueba.
* Diseñar el entorno de prueba y las herramientas necesarias.
* Capturar la trazabilidad bidireccional entre la base de prueba, las condiciones de prueba, los casos de prueba y los procedimientos de prueba.

### Implementación de la prueba

Durante la implementación de la prueba, se crean y/o se completan los productos de prueba necesarios para la ejecución de la prueba, incluyendo la secuenciación de los casos de prueba en procedimientos de prueba. Por lo tanto, el diseño de la prueba responde a la pregunta "cómo probar", mientras que la implementación de la prueba responde a la pregunta "¿está todo preparado para realizar la prueba?”.

## Trazabilidad entre la Base de Prueba y los Productos de Trabajo de la Prueba

### **Ejecución de la prueba** Como se deben imaginar, es la parte donde damos comienzo a los juegos de pruebas previamente armados. La lista es larga, pero lo resolveremos en 4 etapas principales:

1. Ejecutar pruebas
2. Comparar resultados
3. Analizar y reportar defectos
4. Generar trazabilidad

### Compleción de la prueba

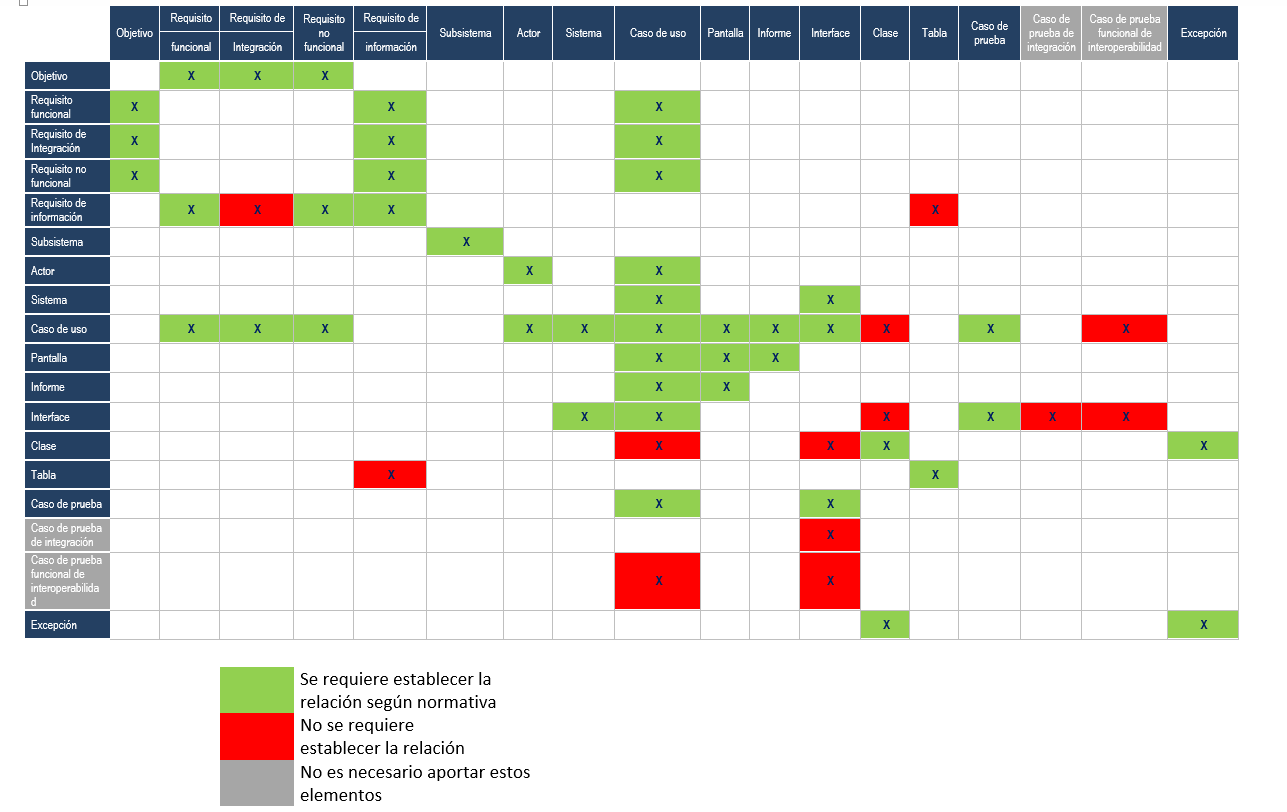
Las actividades de compleción de la prueba recopilan datos de las actividades de prueba completadas para consolidar la experiencia, los productos de prueba y cualquier otra información relevante. Estas actividades ocurren en hitos del proyecto, acorde al proceso de desarrollo que se usa.

### Trazabilidad

Para implementar una monitorización y control efectivos de la prueba, es importante establecer y mantener la trazabilidad a lo largo del proceso de prueba entre cada elemento de la base de prueba y los diversos productos de trabajo de prueba asociados con ese elemento, tal como se describió anteriormente.

* Analizar el impacto de los cambios.
* Hacer que la prueba pueda ser auditada.
* Cumplimiento de los criterios de gobernanza de TI.
* Mejorar la comprensión de los informes de avance de la prueba y de los informes resumen de prueba para incluir el estado de los elementos de la base de prueba.
* Relacionar los aspectos técnicos de la prueba con los implicados en términos que éstos puedan entender.
* Aportar información para evaluar la calidad de los productos, la capacidad de los procesos y el avance de los proyectos en relación con los objetivos de negocio.

#### Ejemplo Trazabilidad



Es un ejemplo que sólo nos sirve para guiarnos.

# La Psicología del Proceso de Prueba

Un elemento de la psicología humana llamado sesgo de confirmación puede dificultar la aceptación de información que esté en desacuerdo con las creencias actuales.

## Trabajando sobre el sesgo de confirmación

Como resultado de esto, algunas personas pueden percibir al proceso de prueba como una actividad destructiva. Para tratar de reducir estas percepciones, la información sobre defectos y fallos debe ser comunicada de manera constructiva. De esta forma, se pueden reducir las tensiones entre las distintas partes del ciclo de desarrollo.

## Sobre la comunicación

Quien se desempeña como tester necesita tener **buenas competencias interpersonales** para poder comunicarse eficazmente sobre defectos, fallos, resultados de la prueba, avance de la prueba y riesgos, y para construir relaciones positivas con las personas que forman parte de su entorno de trabajo.

## Algunas características

* Colaborar, nunca pelear.
* Hacer énfasis en los beneficios de las pruebas.
* Comunicar los resultados de una forma objetiva y neutral.
* Empatizar como recibe la información la contraparte.
* Confirmar que la información fue transmitida correctamente.

## Formas de Pensar del Probador y del Desarrollador

### Mentalidad tester

La mentalidad de un tester debe incluir curiosidad, pesimismo

profesional, sentido crítico, atención al detalle y motivación para una comunicación y relaciones buenas y positivas. Mentalidad que tiende a ampliarse y madurar a medida que se adquiere experiencia.

### Mentalidad desarrollador

La mentalidad de un desarrollador puede incluir algunos de los elementos de la mentalidad de un probador, pero los desarrolladores de éxito, a menudo, están más interesados en el diseño y la construcción de soluciones que en la contemplación de lo que podría estar mal en esas soluciones. Además, el sesgo de confirmación hace difícil encontrar errores en su propio trabajo.

# Reportar resultados

**Cerrando el ciclo**

La presentación de resultados es una de las últimas responsabilidades del tester dentro del ciclo de desarrollo.

### Importancia del reporte

Existen múltiples tipos de reportes para presentar, ya hablamos que varios son medianamente informales, ya que van dedicados al propio equipo, pero también existen otros reportes que están dirigidos a otras partes de la compañía o clientes con mayor grado de criticidad.

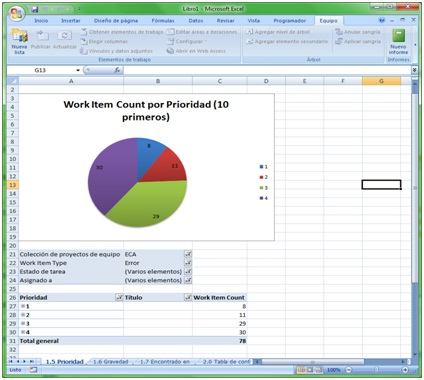
### Reporte de lanzamiento

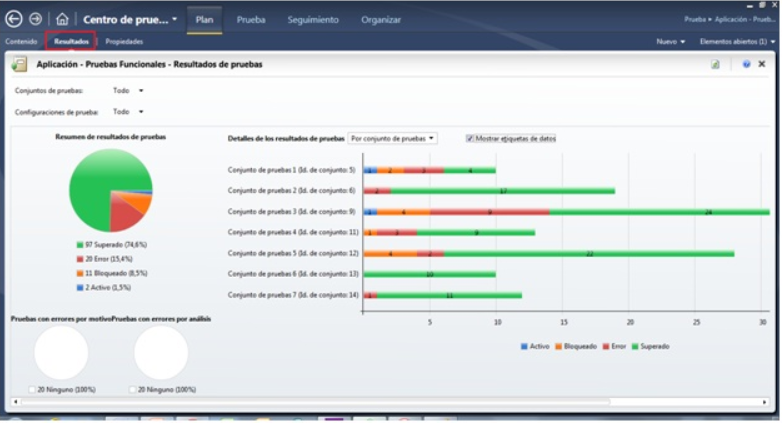
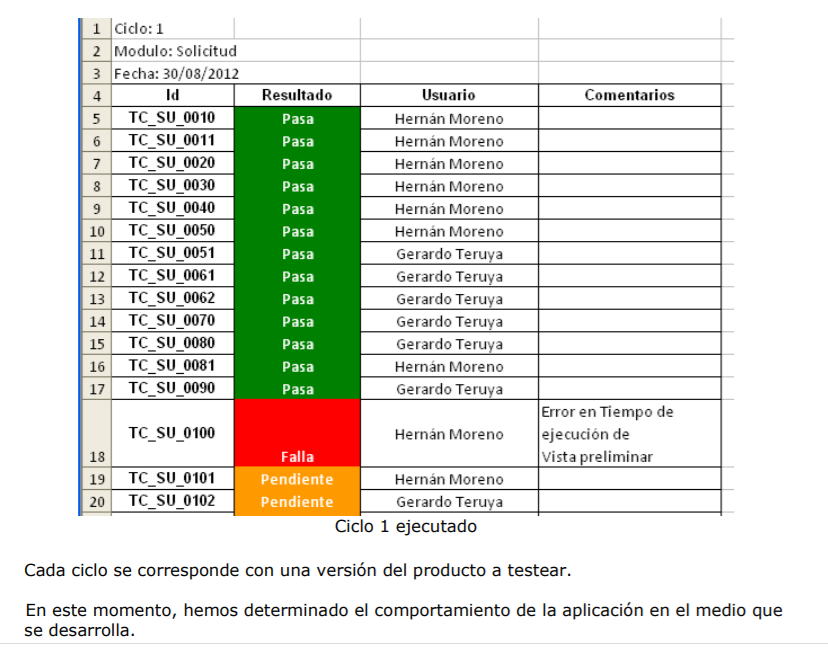
Una vez finalizada la iteración entre testers y desarrolladores por el producto, o el calendario de pruebas cero, se realiza un último reporte donde indicaremos un detalle final de todo lo completo. Acá incluimos la cantidad de casos de pruebas resueltos sobre el total, la cantidad de bugs encontrados, tiempo que se utilizó para realizar esto, y muchas cosas más.

### Criterio del QA

Al momento de presentar el reporte de lanzamiento de versionado, es importante siempre dejar un comentario como profesionales, si nosotros creemos que el proyecto debería ser lanzado a producción o no. Esta tarea será llevada adelante basada en la severidad de los defectos sin resolver, la cantidad de los mismos, o incluso notas de usabilidad que creemos que son necesarias.

### Ejemplo de reporte





# Puesta en producción y monitoreo

### Últimas etapas

Si bien ya dijimos que el reporte de resultados, es la última tarea del tester, hay un par de pasos previos a llegar a esa estadía del producto.

Es decir, si debemos enumerar las últimas etapas de puesta en marcha de un producto o versión, hay que separarlas en 4 partes.

### Partes de la última etapa

1. UAT
2. Reporte
3. Pruebas en producción
4. Monitoreo

#### UAT- Testing de aceptación de usuario

Hay que remarcar la importancia de esta parte en las últimas etapas del ciclo de desarrollo.

En el UAT nos centramos en hacer pruebas en la última versión que será entregada a producción.

##### Defectos en UAT

En muchas circunstancias se encuentran nuevos defectos en la etapa de UAT y estos deberán ser cargados como cualquier otra etapa del testing. Es importante remarcar que una vez que el tester apruebe este UAT para salir a producción deberá realizarse el reporte de resultados.

#### Reporte de resultados

##### Presentación reporte

El reporte deberá indicar múltiples datos a tener en cuenta para salir a producción, y esto se realizará posteriormente del ciclo de UAT. La importancia y relación de ambas es que el tester indicará en el reporte, cuál fue la versión de UAT que se aprueba para poder salir a producción.

#### Pruebas en producción

Cuando se pacta, con fecha y hora, la salida a producción de algo nuevo, el tester deberá hacer pruebas con el producto puesto en producción para asegurar que todo funciona como esperado, sin ninguna anomalía o fallo productivo.

#### HOTFIX

Existen casos donde al subir algo a producción, el programa presenta defectos. Para estos casos, los desarrolladores implementan cambios en producción muy rápidos, para solucionar el problema. Esto se lo llama HOTFIX

y también será una prueba en producción.

#### **Rollback**

En otros casos, la nueva versión subida puede presentar defectos y fallas de mayor gravedad, debido a problemas de compatibilidad o integración en producción. Frente a estos casos más críticos, se suele realizar rollback que es volver a la versión anterior del sistema en producción.

#### MONITOREO

##### AUTOMATIZACIÓN EN LA CANCHA

El monitoreo de producción que conocemos como prueba de mantenimiento, refiere a aquello que hacemos para tener seguridad de que el sistema es estable en producción. Acá juegan las pruebas automatizadas, que se pueden programar para correr múltiples veces en el día, para asegurar la integridad del producto.

# Comunicación con el equipo

### Relación con desarrolladores

Como ya se explicó, la relación con los desarrolladores es clave para poder lograr un mejor producto, ya que el ciclo de intercambio de producto que se da con ellos, es el más recurrente de todo el circuito.

Es importante mantener una buena relación con ellos en pos de poder lograr mejor entendimiento de las cosas que puedan ir sucediendo.

#### Presentando evidencia

Un problema muy recurrente en la relación con los devs es la falta de evidencia de algún defecto.

Es primordial a la hora de la presentación de bugs, contar con toda la evidencia necesaria para el desarrollador. Algunos errores no pueden repetirlos de forma natural y por eso mismo, el desarrollador va a necesitar la mayor cantidad de información de nuestra parte para solucionarlo.

### Relación con infraestructura-DevOps-Sysadmins

De acuerdo a la empresa en la que se encuentren, poseen muchos nombres, pero su relación con nosotros suele basarse en el armado de ambientes de prueba. Son quienes mantienen la salud de los ambientes y servidores en el nivel de equipos, y es a quienes debemos recurrir para buscar problemas de conectividad y otras posibles.

#### ¿Problemas de server?

Aunque no sea algo normal, suelen levantarse bugs también al equipo de infraestructura, cuando tenemos pruebas que demuestran lentitud o pérdida de paquetes, como alguno de los errores más comunes. Estos se podrán encontrar a través de pruebas no-funcionales.

### Relación con Product Owners

Los product owners (a partir de ahora PO), son los encargados de guiar el producto hacia el lado que el negocio quiere que se dirija. Además, son los proveedores de la primera documentación que vamos a trabajar, con la información que agregaremos al proyecto.

#### Dueños del proyecto

Los PO dirigen el proyecto y son los responsables por guiar el producto hacia el norte que ellos creen o que el negocio pide. Desde nuestro lado, como agentes de calidad, siempre podemos hacer sugerencias para mejorar el producto, discutir con el PO sobre ellos y quizá lograr sumar estas cosas como mejoras al proyecto.

### Relación con diseñadores

Los diseñadores arman mocks o “simulacros” de cómo se debe ver la plataforma trabajada. Es común que las primeras veces, no encontramos detalles que solo el ojo de un diseñador podrá, pero lo importante es contar con ellos como aliados para agudizar nuestra mirada en los diseños implementados.

#### Importancia del mock

Es importante que el diseño final respete el mock, ya que esto es lo diseñado y aprobado por el negocio. Es nuestra responsabilidad asegurar que el producto final sea lo más fiel posible al mock y velar por ello. Los diseñadores pueden ayudar a revisar o consultores cuando tenemos dudas.

### Relación con otros roles externos

El tester suele estar en el centro de todos los proyectos, por eso cada rol que haya dentro del proyecto, tiene que ser un aliado para nuestro trabajo.

Si bien destacamos los cuatro más importantes, podemos encontrarnos trabajando con gente de seguridad informática, arquitectos, especialistas en base de datos, etc...

### CONOCIMIENTO TÉCNICO

Muchas veces los niveles de tecnicismos pueden sobrepasar nuestro conocimiento y no es algo para sentirse mal. Nunca tengamos miedo de preguntar a los especialistas para que nos explican.

El QA es uno de las especialidades que cuanto más sabe de los trabajos ajenos, mejor puede hacer el propio. Y eso es parte de nuestra responsabilidad como calidad.

#### **¿QUIERES SABER MÁS? TE DEJAMOS MATERIAL AMPLIADO DE LA CLASE**

* [Documento ISTQB en Español - Página 25 a 35](https://141647fc-7a5b-4407-a84a-78567bb3f5b0.filesusr.com/ugd/acfdb9_743bf3d31dde49578c94d97e5b96b9da.pdf)

#DEMOCRATIZANDOLAEDUCACIÓN