

PIRÁMIDE DEL TESTING

Universidad Tecnológica Nacional - Facultad Regional Córdoba

Ingeniería y Calidad de Software

Grupo N° 6 - Alumnos: Marc Florencia, Luisina Capovilla, Ezequiel Luzara

INTRODUCCIÓN

Tradicionalmente el testeo de software se ha realizado desplegando la app en algún ambiente de pruebas y ejecutando manualmente una serie de casos pruebas previamente definidos en un script de pruebas.

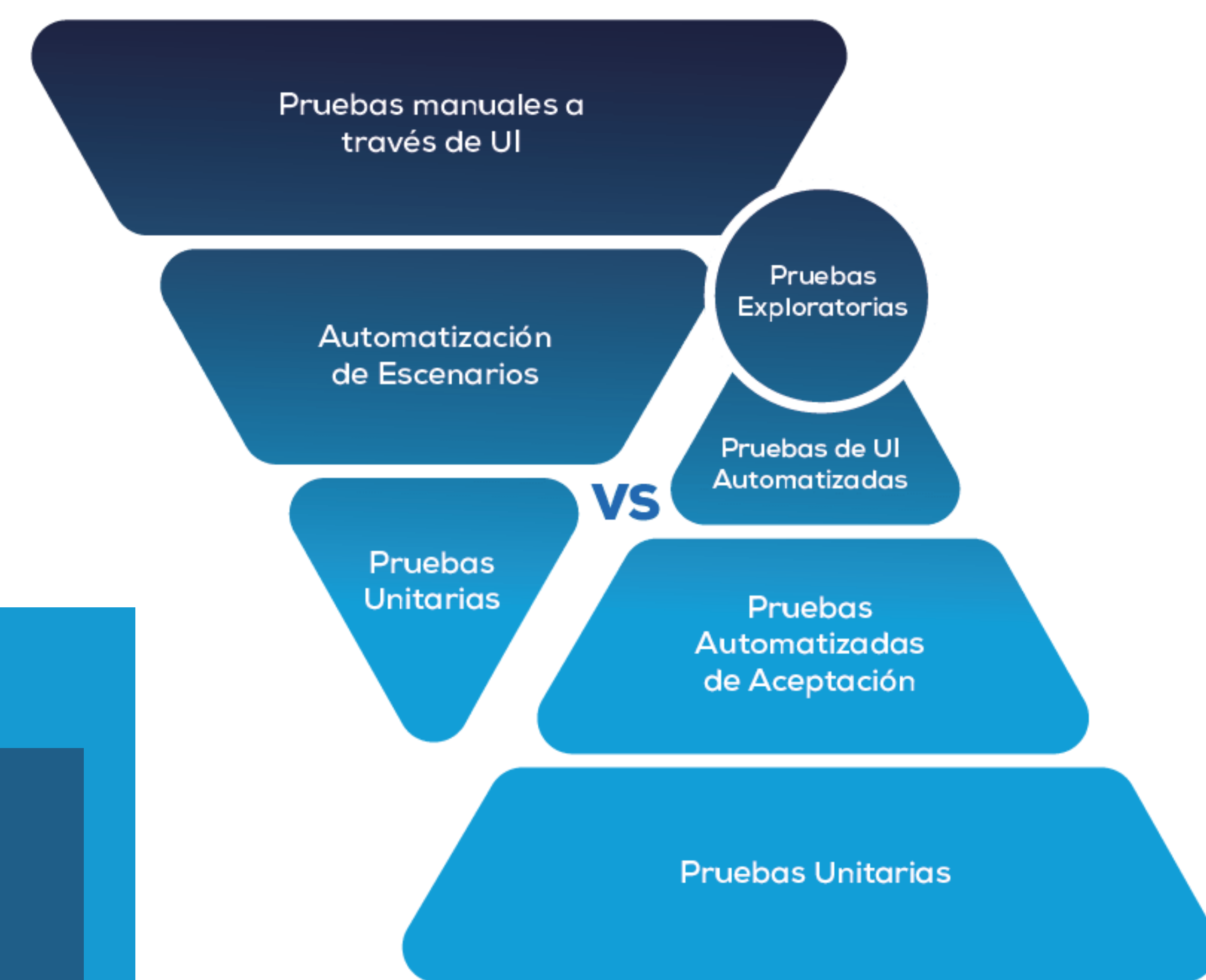
Una de las herramientas para un correcto desarrollo de las pruebas automatizadas fue presentada por Mike Cohn “La Pirámide de Pruebas Automatizadas” que sirve para explicar las diferencias del Software Testing trabajando con metodologías convencionales y trabajando con metodologías ágiles o de forma iterativa.

AMBIENTES TRADICIONALES VS AMBIENTES ÁGILES

En la forma tradicional (que es la pirámide de la izquierda) la mayoría de las pruebas son manuales y si bien hay un grado de automatización, es pequeño. Las pruebas unitarias son las que son llevadas adelante por los desarrolladores. Aquí el foco es encontrar errores.

En cambio, la pirámide que representa cómo se ejecuta el testing en ágil (la de la derecha) nos muestra que la mayoría de las pruebas son automatizadas. Las pruebas funcionales manuales se tratan de reducir al mínimo. Aquí el foco es prevenir errores.

Tradicional (Encontrar errores)



Agiles (Evitar Errores)

PRUEBAS DE UI:

Se realizan para garantizar que la aplicación funcione correctamente de un extremo a otro.

La mayoría de estas pruebas se llevan a cabo utilizando la propia interfaz de usuario de la aplicación y constituyen la capa superior de la pirámide de pruebas: **el 10% de todos los casos de prueba previstos**. Si bien estas pruebas ofrecen una cobertura más amplia, también son más lentas y frágiles.

Por lo tanto, se priorizan los escenarios críticos y los flujos clave para asegurar una retroalimentación rápida y efectiva.



PRUEBAS UNITARIAS:

Se automatizan las pruebas a nivel de unidades individuales de código, como funciones o métodos. Estas pruebas son rápidas de ejecutar y se centran en verificar que cada unidad funcione correctamente.

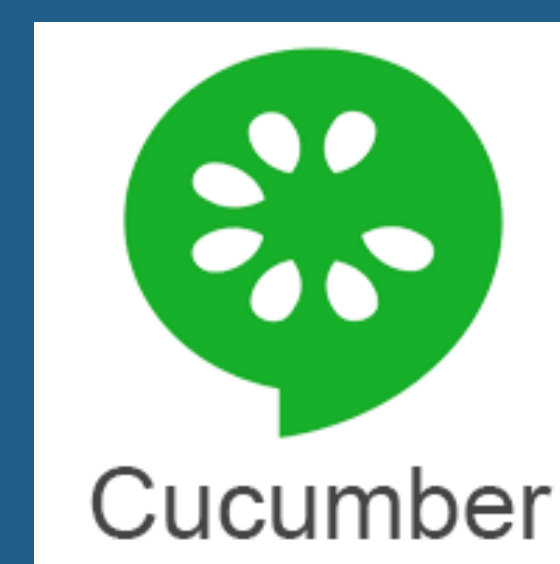
Son útiles para atrapar errores tempranos y garantizar que los componentes básicos del software sean sólidos.

Deben ocupar aproximadamente el 70% de todos los casos de prueba que pretende ejecutar.



PRUEBAS DE INTEGRACIÓN Y ACEPTACIÓN:

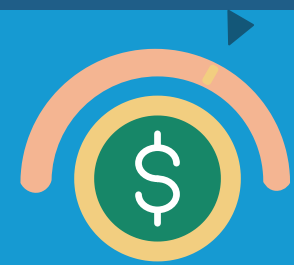
Garantizan que la integración entre varios componentes de la aplicación esté funcionando correctamente. Son en su mayoría pruebas funcionales basadas en API, que constituyen la capa intermedia de la pirámide de pruebas y **constan de el 20%** de todos los casos de prueba. El uso de pruebas de integración debe ser limitado o usarse con mucha menos frecuencia que las pruebas unitarias.



PRUEBAS EXPLORATORIAS:

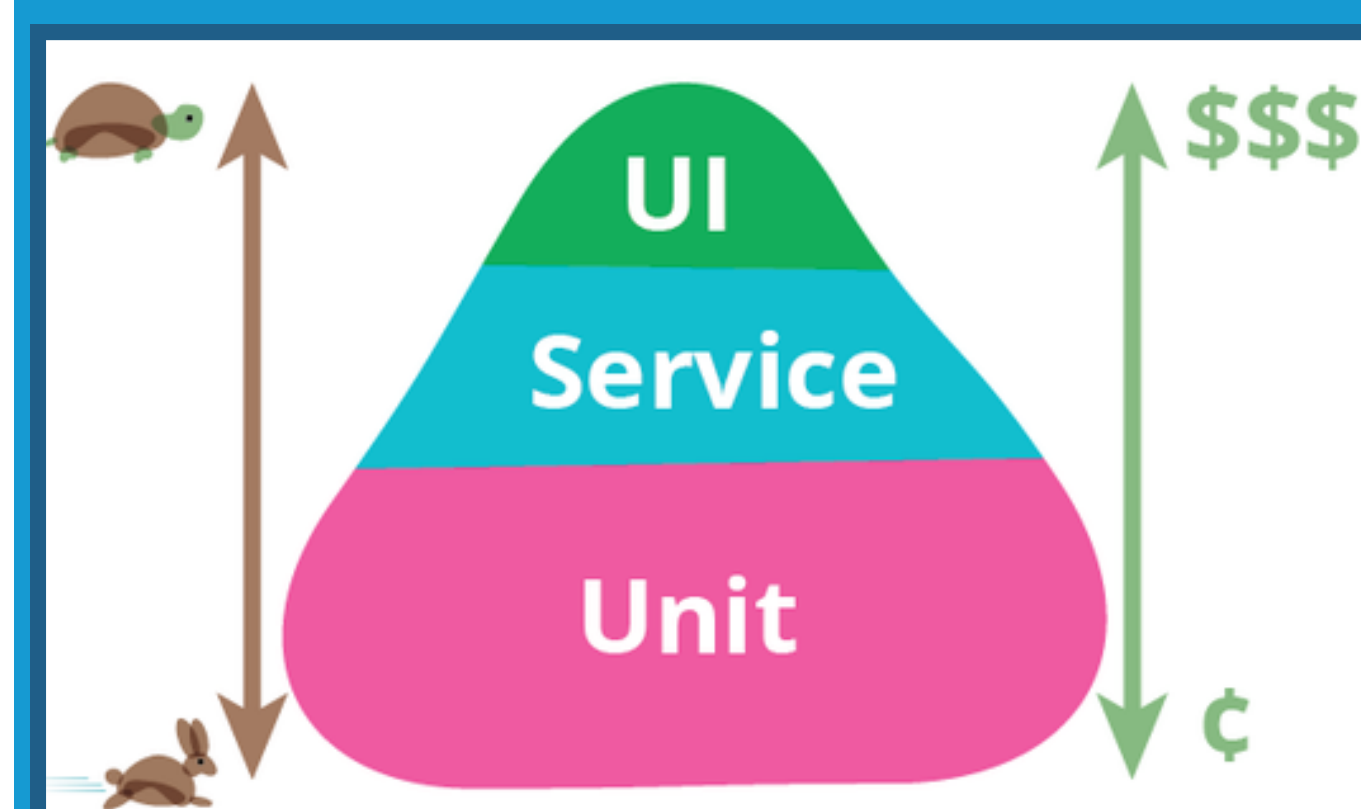


Abarcan la exploración humana y las pruebas manuales que no son fácilmente automatizables. El testing manual implica la evaluación cuidadosa y la interacción directa de los testers con la aplicación para identificar problemas complejos, flujos poco comunes y aspectos de usabilidad que no se pueden capturar fácilmente mediante pruebas automatizadas.



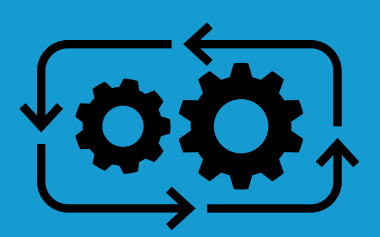
EL COSTO DE LOS TESTS

Los tests más rápidos de implementar son las pruebas unitarias, esto hace que sean las de menor coste, pero también son las que menos bugs podrían ayudarnos a encontrar. Mientras que las pruebas UI no darían mucha más confianza pero el costo de implementación y mantenimiento es muy alto.



También sirve para representar el tiempo/esfuerzo que conlleva cada capa así como el coste de los mismos, siendo estos inversamente proporcionales: es decir, los tests unitarios son rápidos y baratos de hacer y los de interfaz de usuario requieren más tiempo y dinero.

AUTOMATIZACIÓN

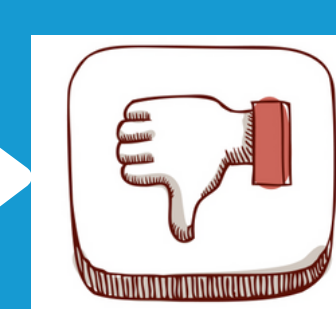


VENTAJAS

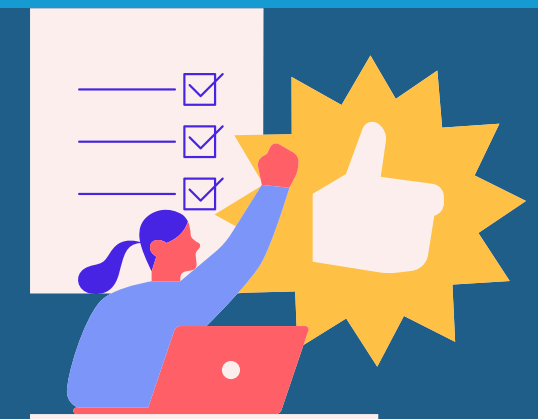


- Mejora de la calidad de producto
- Ser competitivos frente al cliente.
- Mayor homogeneidad.
- Reducción del coste de la mano de obra.
- Eliminación de los trabajos rutinarios.

DESVENTAJAS



- Lentitud y dificultad de adaptación a los cambios de un proceso ya automatizado para producir otros modelos diferentes.
- Requiere especialización por parte del personal.
- La inversión en equipos puede ser más costosa.



CONCLUSIÓN:

En resumen, la Pirámide de Automatización de Pruebas equilibra la cobertura de pruebas y la eficiencia en el proceso de desarrollo de software. La adición de la capa de pruebas manuales y exploratorias en la cima de la pirámide refleja la importancia de la interacción humana en la detección de problemas sutiles en la evaluación de la experiencia del usuario.

REFERENCIAS:

- <https://medium.com/nिकासource/QU%C3%A9-ES-LA-PIR%C3%A1MIDE-DE-PRUEBAS-6AC4356C963F>
- <https://josepablosarco.wordpress.com/2014/09/09/testing-automatizado-en-agile/>
- LIBRO: COHN MIKE - AGILE TESTING A PRACTICAL GUIDE FOR TESTER AND AGILE TEAMS - EDITORIAL ADDISON WELSEY - AÑO 2009
- <https://cl.abstracta.us/blog/PIRAMIDE-DE-AUTOMATIZACION/>
- <https://academy.pega.com/topic/test-pyramid/v1>
- <https://blog.gloin.es/qaless-manual-de-testing-para-desarrolladores/>

AGRADECIMIENTOS:

