

1. Verdadero y falso

- a. V.
- b. F. Se diferencian en que regresion hace referencia asegurar que los nuevos features no hagan que los “viejos” fallen, un nuevo cambio no puede romper algo que ya estaba andando. En cambio Cobertura hace referencia al porcentaje de código desarrollado que está incluido (cubierto) por los tests.
- c. F. El porcentaje de cobertura no es un número fijo, este depende de muchos factores, entre estos factores está: el tamaño de la empresa, su infraestructura, a que se dedica la empresa, etc. Sin embargo, se suele recomendar que la cobertura esté entre un 80% y 90%. Lo importante es que la cobertura sea relevante.
- d. V.
- e. F. Cobertura no significa usabilidad. Tener un alto nivel de cobertura significa que gran parte del código desarrollado es testeado, esto nos permite tener un mayor grado de seguridad de que lo que uno desarrolló funciona. Sin embargo, alta cobertura no significa que el código esté libre de errores, estos siempre pueden ocurrir. Por otro lado, la usabilidad tiene que ver con cosas como: la claridad, simpleza o el estilo del producto.
- f. F. El beneficio del ROI como consecuencia de aplicar Tests de cobertura tiene que ver con que esta permite eliminar errores en épocas tempranas del desarrollo, lo que ahorra dinero (porque arreglarlo en estas etapas tempranas es más barato), problemas y discrepancias con las especificaciones (las que se pueden arreglar de forma más sencilla en las primeras etapas).
- g. F. La implementación de test de cobertura, como dije anteriormente, no aseguran encontrar TODOS los errores, algunos de ellos, como por ejemplo los edge cases, puede que errores asociados a ellos no sean detectados. Toda técnica es susceptible a fallar, incluidos los test de cobertura.
- h. F. El statement coverage evalúa solamente la ejecución de todos los condicionantes del programa. El path coverage se encarga de testear todos los posibles caminos.
- i. F. Decision / Branch coverage evalúa que se ejecute todas las posibles aristas de un código, tomando la cantidad mínima de caminos posibles.
- j. V.

2. Si - No - No aplica

- a. Si - El sitio posee un menú de navegación persistente en todas las páginas que permite con pocos clicks ir a la sección del mismo que desee. Además, en ese menú posee un botón de home en forma del logotipo de mercado libre.
- b. Si - Cada sección presenta una forma muy intuitiva de leer la información y de navegar a la sección pertinente para el uso que cada uno necesite.
- c. Si - Las opciones se presentan orientadas (de forma clara) a la compra de artículos de diferentes formas, como por ejemplo, por

ofertas, artículos ya comprados, por rubro al que pertenezca el artículo, etc.

- d. Si - El sistema de navegación es amplio, con muchos ítems en vez de submenús, a excepción de la categoría de tecnología. Esta sección presenta más ítems dentro ya que es la categoría más comprada de mercado libre y de la que dispone mayor variedad.
- e. Si - La estructura que lleva a un usuario a realizar la compra es extremadamente simple y sin niveles innecesarios. Esto se evidencia en el acto de realizar una compra, al tener muchos datos ya cargados (como la dirección de entrega y los datos de la tarjeta), la página no vuelve a pedirlos, solo solicita su confirmación. Esto agiliza el proceso y lo simplifica.
- f. No - Mercadolibre se enfoca en facilitar la compra de un artículo, pero no se puede ingresar a la sección de seguridad de la cuenta o de direcciones cargadas (secciones de configuración), para esto es necesario pasar por la sección de perfil primero. Si uno se enfoca en la navegabilidad entre categorías de productos, este punto si se cumple.
- g. NA - No aplica ya que en el caso de Mercadolibre, no tiene sentido tener pestañas ya que el usuario se focaliza normalmente en 1 solo producto o acción por vez. Si necesita seleccionar más de un producto tiene el carro de compras o favoritos para almacenar todo artículo de interés.
- h. No - No existe un mapa del sitio a simple vista o de fácil acceso. Alejandro encontró uno pero en Google y este estaba desactualizado, lo que quiere decir que este punto no se cumple y que esa no es una feature que le importe mantener a Mercadolibre. Está, pero es "inútil".
- i. No - El mapa del sitio no está en todas las paginas, es una página aparte.
- j. Si - Cada página posee en la parte superior un título grande que dice en qué parte del sitio se encuentra el usuario o muestra algún tipo de imagen que lo ilustra, como por ejemplo la sección de vehículos que muestra imágenes e iconos que simbolizan "vehículo". Sin embargo, la sección de supermercado no presenta ningún tipo de indicación a simple vista que le indique al usuario que se encuentra en esa parte. Uno puede intuir que está en ella porque está organizada de forma que parece una revista de promociones de supermercados.
- k. Si - Las etiquetas de las categorías describen siempre el tipo de producto que uno está por comprar o que va a haber un mix de productos bajo una misma categoría acorde.
- l. IDEM k
- m. Si - Las etiquetas son muy descriptivas y utilizan las palabras adecuadas para entender el propósito de la misma.
- n. Si - El sitio mantiene un mismo estándar a lo largo de todas sus páginas, ya que facilita el uso a usuarios nuevos o no experimentados.

- o. Si - Mantienen consistencia en todas las páginas de lo que se entiende por enlace, lo que facilita su uso a usuarios no experimentados.
3. Importancia de Staging para los siguientes profesionales:
  - a. **Desarrollador:** Es importante para el desarrollador ya que le permite poder desarrollar features individualmente y posteriormente integrarlas con los desarrollos de sus compañeros. Su entorno es más cambiante y permite aislarlo de los demás. Además, los errores que surgen inevitablemente del desarrollo, no suceden con datos importantes para los usuarios o durante el uso de la aplicación por parte del mismo.
  - b. **QA Engineer:** Porque les permite poder realizar su labor en un entorno que es similar al real (producción), pudiendo testear de forma más confiable el comportamiento del sistema en esas situaciones. Su entorno necesita mayor estabilidad, por ejemplo respecto a la compatibilidad de versiones con dependencias. Al tener su entorno aislado del de desarrollo, el QA puede probar libremente sin preocuparse por esos cambios dentro de un mismo release. Se preocupara de esas cuestiones cuando traiga a su entorno lo desarrollado en el entorno de desarrollo. Aisla problemas de compatibilidad por ejemplo.
  - c. **Usuario final:** Al usuario final le asegura que el software ha sido testeado en diferentes situaciones múltiples veces, entregando un producto con una probabilidad baja (no nula) de presentar errores. Además, el pasar por diferentes entornos es pasar por diferentes filtros. Un producto que es revisado o filtrado múltiples veces es de mayor calidad. Se pueden evitar situaciones como por ejemplo: elementos en una aplicación que no siguen el estándar o modelo planteado en el diseño gráfico. Se aíslan los datos de producción (que son los que usa el usuario) de los datos de desarrollo, los cuales se pueden corromper o modificar no intencionadamente.
4. El equipo de calidad debería tener autonomía de Deployment para la ejecución de pruebas en el Stage 4 (QA) porque de esa manera se permite mantener un entorno parecido a producción pero que no es, y además está separado del de desarrollo que es un entorno muy cambiante y con muchos errores.
5. Las OKRs (Objective and Key Results) son un framework, metodología o paradigma de objetivos o desafíos impuestos por equipos y personas para lograr crecimiento y mejora en diferentes ámbitos, como el personal, el laboral, el económico y el profesional. No son acciones, sino los resultados de las mismas. Uno define un objetivo (metas y aspiracionales) y sus resultados clave (métricas desafiantes y evaluables con una fecha límite, que sirven para saber si se está cumpliendo el objetivo), para mantener la organización orientada a un aspecto de interés.

Objetivo: 2023: Mejorar la generación de documentación del testing.

KR1: Para comienzos de Febrero utilizar el tablero kanban para registrar el 80% de las actividades realizadas.

KR2: Describir cómo replicar un error en el 90% de los test manuales fallidos, para fines del 3Q.

KR3: Capacitar a la totalidad del personal en la generación de test y su mantenimiento para fines del 2Q.