**Organización de la prueba**

Al momento de desarrollar un software nos daremos cuenta de que cuando lo finalicemos, en realidad el trabajo no acaba ahí. Llegará el día en el que el proyecto será publicado y que el cliente lo empiece a usar. Todo debería funcionar bien, pero ¿qué pasa si luego se debe realizar un cambio imprevisto o si se visualiza un error que necesita ser corregido? No podemos hacer los cambios directamente sobre el ambiente en el que el cliente está utilizando el software, porque podríamos romperlo y dejarlo inoperativo. Es por eso que deben existir diferentes ambientes de trabajo, donde se pueda desarrollar y probar los cambios antes de que llegue al ambiente del cliente.   
Entendemos como ambiente de trabajo al entorno con todos los recursos necesarios para que se pueda ejecutar un sistema. En esta parte de la materia, aprenderemos cómo se organizan las diferentes pruebas a lo largo de los ambientes, entendiendo qué pruebas utilizar y dónde deben ejecutarse.

# Ambientes de trabajo

### ¿Qué es un ambiente?

La ejecución de las pruebas se realiza en diferentes espacios de trabajo de acuerdo a la etapa del ciclo de desarrollo en el que se encuentre el sistema en construcción o mantenimiento. Estos entornos son conocidos como ambientes. En otras palabras, hacen referencia a un servidor con ciertos recursos asignados, software y librerías instalados, su propia base de datos y una configuración determinada.

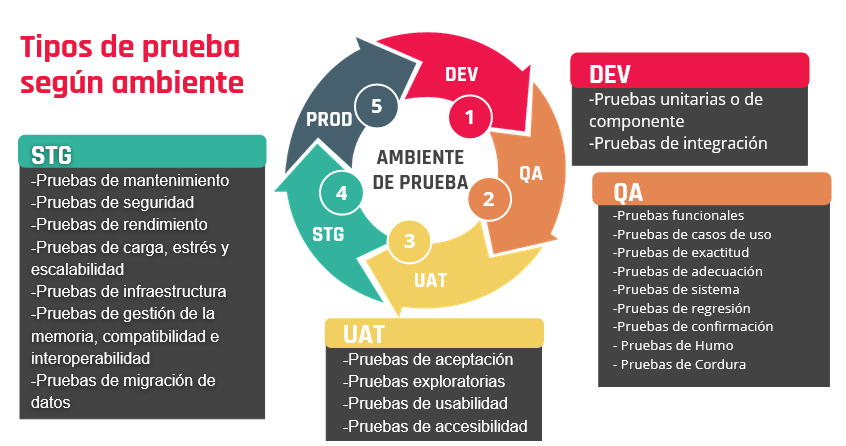
Esto nos permitiría desarrollar aplicaciones de forma segura y con entornos diferenciados para realizar la programación, realizar pruebas, compartir resultados con los clientes y permitirles realizar pruebas y prácticas; y finalmente publicar una aplicación robusta y estable.

Cada uno de los ambientes son utilizados con un fin específico y presentan ciertas ventajas sobre los otros en determinado momento del proceso de trabajo.

### Niveles de ambientes

Entonces, podemos decir que es conveniente distinguir los siguientes entornos:

* **a) Ambiente de desarrollo o DEV** :este entorno es el espacio de trabajo donde el programador desarrolla el código de la aplicación, realiza pruebas iniciales y comprueba si la aplicación se ejecuta correctamente con ese código. Este ambiente puede ser local o en la nube, de acuerdo a la necesidad del proyecto.
* **b) Ambiente de pruebas o QA** : el entorno de pruebas suele estar ubicado en un servidor en la nube o en una granja de servidores locales (laboratorio). Permite minimizar incidencias en etapas posteriores, ya que el tester ejecutaría las primeras pruebas de funcionalidad en este ambiente.
* **c) Ambiente de UAT**: el entorno de UAT (o de pruebas de aceptación de usuario) permite a los usuarios del cliente poder verificar que los cambios realizados son los que realmente se solicitaron, evaluando a su vez accesibilidad y usabilidad.
* **d) Ambiente de preproducción o STAGE** :este entorno debería poseer una configuración técnica idéntica a la que nos encontraremos en el entorno de producción. El propósito principal de este entorno es emular al entorno de producción con el fin de probar las actualizaciones y asegurar que estas no corromperán la aplicación en los servidores en producción cuando sean desplegadas. De esta forma se minimizan las caídas del sistema y corte de los servicios en producción.
* **e) Ambiente de producción o PROD** :este es el entorno donde finalmente se ejecuta la aplicación, donde acceden los usuarios finales y donde se trabaja con los datos reales de negocio. Es un servidor que posee las mismas características y configuración que tendrá el servidor de preproducción. Aunque, en este caso, puede estar configurado por más de un servidor, para efectos de balanceo de carga en aplicaciones que requieren una infraestructura con capacidad de manejar un tráfico de usuarios pesado y miles de conexiones concurrentes.

**Pruebas según el ambiente**

En las siguientes pantallas no desarrollaremos el **ambiente PROD** debido a que:

* En general, los probadores no tienen acceso a este ambiente.
* En el caso de tener acceso y realizar pruebas:
  + No se deben realizar acciones que generen datos.
  + Se corre el riesgo de ingresar datos basura.
  + Se interfiere en los datos de seguimiento.

**DEV**

**Pruebas unitarias o de componente:** También se conocen como pruebas de módulo. Se centra en los componentes que se pueden probar por separado. Tiene como objetivo encontrar defectos en el componente y verificar que los comportamientos funcionales y no funcionales del componente son los diseñados  y especificados.

**Pruebas de Integración:** Se centra en las interacciones entre componentes o sistemas. Los objetivos de la prueba de integración incluyen encontrar defectos en las propias interfaces o dentro de los componentes o sistemas y verificar que los comportamientos funcionales y no funcionales de las interfaces sean los diseñados y especificados.

**QA**

**Pruebas funcionales:** Incluye pruebas que evalúan las funciones que el sistema debe realizar. Los requisitos funcionales pueden estar descritos en productos de trabajo tales como especificaciones de  requisitos de negocio, épicas, historias de usuario, casos de uso y especificaciones funcionales. También pueden  estar sin documentar.

**Pruebas de casos de uso:** Proporcionan pruebas transaccionales, basadas en escenarios, que deberían emular el uso del sistema.

**Pruebas de exactitud:** Comprenden el cumplimiento por parte de la aplicación de los requisitos especificados o implícitos y también puede abarcar la exactitud de cálculo.

**Pruebas de adecuación:** Implican evaluar y validar la eficiencia de un conjunto de funciones para la consecución de las tareas especificadas previstas. Estas pruebas pueden basarse en casos de uso.

**Pruebas de sistema:** Se centra en el comportamiento y las capacidades de todo un sistema o producto, a  menudo teniendo en cuenta las tareas extremo a extremo que el sistema puede realizar y los  comportamientos no funcionales que exhibe mientras realiza esas tareas.

**Pruebas de regresión:** Implican la  realización de pruebas para detectar efectos secundarios no deseados, luego de cambios hechos en una parte del código que puedan afectar accidentalmente el comportamiento de otras partes del código.

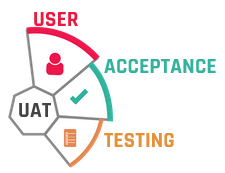
**Pruebas de confirmación:** Consiste en volver a ejecutar los pasos para reproducir el fallo o los fallos causados por un defecto en la nueva versión de software, una vez corregido el defecto, para así confirmar que el defecto original se ha solucionado satisfactoriamente o detectar efectos secundarios no deseados.



**Pruebas de cordura:** Es una prueba de regresión acotada que se centra en una o unas pocas áreas de funcionalidad. Se utiliza para determinar si una pequeña sección de la aplicación sigue funcionando después de un cambio menor.

**Pruebas de humo:** Se lleva a cabo un smoke test para asegurar si las funciones más importantes de un programa están trabajando correctamente, pero sin molestarse con los detalles más finos.

**UAT**

**Pruebas de aceptación:** Se centra normalmente en el comportamiento y las capacidades de todo un sistema o producto. Además, pueden producir información para evaluar el grado de preparación del sistema para su despliegue y uso por parte del cliente (usuario final).

**Pruebas exploratorias:** Se diseñan, ejecutan, registran y evalúan de forma dinámica pruebas informales (no predefinidas) durante la ejecución de la prueba. Los resultados de la prueba se utilizan con el objetivo de aprender más sobre el componente o sistema y crear pruebas para las áreas que pueden necesitar ser probadas con mayor intensidad.

**Pruebas de usabilidad:** Evalúan la facilidad con la que los usuarios pueden utilizar o aprender a utilizar el sistema para lograr un objetivo específico en un contexto dado.

**Pruebas de accesibilidad:** Incluyen y evalúan la accesibilidad que presenta un software para aquellos con necesidades particulares o restricciones para su uso. Esto incluye a aquellos usuarios con discapacidades.

**STG**

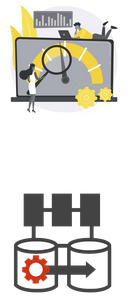
**Pruebas de mantenimiento:** Se centra en probar los cambios en el sistema, así como en probar las piezas no modificadas que podrían haberse visto afectadas por los cambios. El mantenimiento puede incluir lanzamientos planificados y no planificados.

**Pruebas de seguridad:** Las pruebas de seguridad se podrían definir como el conjunto de actividades que se llevan a cabo para encontrar fallas y vulnerabilidades en el sistema, buscando disminuir el impacto de ataques y pérdida de información importante.

**Pruebas de rendimiento:** Se implementan y se ejecutan para evaluar las características relacionadas con el rendimiento del destino de la prueba, como los perfiles de tiempo, el flujo de ejecución, los tiempos de respuesta y la fiabilidad y los límites operativos. También se pueden realizar en STG.

**Pruebas de carga, estrés y escalabilidad:** Una prueba de **carga** garantiza que un sistema pueda controlar un volumen de tráfico esperado. Una prueba de **estrés** es en la que se somete al sistema a condiciones de uso extremas para garantizar su robustez y confiabilidad. Las pruebas de **escalabilidad** garantizan la escalabilidad de un sistema, es decir, que pueda soportar el incremento de demanda en la operación. También se pueden realizar en QA encontrando el correspondiente escalar con respecto a un ambiente de PROD.

**Pruebas de infraestructura:** Incluyen todos los sistemas informáticos internos, los dispositivos externos asociados, las redes de Internet, la nube y las pruebas de virtualización.

**Pruebas de gestión de la memoria:** Evalúan el estado y la integridad de la memoria del sistema para identificar problemas potenciales.

**Pruebas de compatibilidad:** Incluyen las pruebas para comprobar que el sistema es compatible con todos los navegadores de Internet y todos los sistemas operativos del mercado.

**Pruebas de interoperabilidad:** Se refieren a aquellas donde se realiza la evaluación de la correcta integración entre distintos aplicativos, sistemas, servicios o procesos que conforman una plataforma o solución tecnológica.

**Pruebas de migración de datos:** Incluyen las pruebas realizadas al transferir datos entre tipos de dispositivos de almacenamiento, formatos o sistemas de cómputo.