# Reto 1 - Proceso de autenticación

1. Identifica los tipos de casos de prueba que estarían enlazadas a los requerimientos propuestos.

**Rta:** Pruebas que deden realizarse:

* + SmockTest
  + Pruebas Funcionales
  + Pruebas de seguridad
  + Pruebas de integración
  + Pruebas Unitarias

1. Menciona que técnica de diseño de casos de prueba usarías para poder generar dichos casos.

**Rta:** Técnicas:

* + Valores Limites
  + Clases de equivalencia
  + Tablas de decisión
  + Casos de Uso
  + Técnicas combinatorias

1. Identifica, registra y describe los bugs o mejoras que consideres necesarios.

**Rta:**

1. Implementa los tipos de pruebas automatizadas que consideres necesarios (unitarias, servicios, ui , perfomance), de tal manera que se pueda evaluar los requerimientos solicitados. Se debe crear un proyecto en una carpeta aparte llamada qaChallenge (utiliza el framework de automatización que prefieras y deberas subirlo a tu repositorio).

**Rta:**

# Reto 2 - Entornos de trabajo y niveles de pruebas

Entorno de Desarrollo:

* Tipos de pruebas a realizar:
  + Integración
  + Componentes
  + Performance
  + Stress
  + Pruebas funcionales

Entorno de Testing:

* Tipos de pruebas a realizar:
  + Smock Test
  + Funcionales en base a las historias de usuarios, documentación funcional (Evidenciar todo tipo de pruebas)
  + Usabilidad en base a las reglas de la heurística

Entorno de Producción:

* Se debe replicar las pruebas que se realizan en testing
* UAT

***Nota:*** *Antes de realizar las pruebas en el ambiente de Prod, incorportaria el ambiente de homologación, en el cual el usuario podrá realizar las pruebas necesarias, con el fin de brindar mayor aseguramiento a la calidad y mejorar los resultados obtenidos en los otros ambientes. Este ambiente puede ser una replica exacta de lo que pasa en Test para no correr el riesgo de realizar pruebas críticas en Prod.*

# Reto 3 - Priorización de errores

Prioridad de errores según criticidad:

1. Las credenciales de usuario no están cifradas en la base de datos.

**Rta:** Toda credencial debe estar cifrada en base por temas de seguridad, pueden existir personas mal intencionadas y hacerse pasar por otras en un servicio sin seguridad rigurosa.

1. Los precios no varían cuando una transacción se realiza en diferentes monedas.

**Rta:** Los precios de las monedas deben variar dependiendo del tipo de cambio de cada uno de los países desde los cuales pueden accederse al sitio, por otro lado, si la página está orientada a tener una sola moneda de circulación, ej: USD, el precio debería figurar así para los países. Pero como no puedo suponer nada, es algo prioritario.

1. Las transacciones entre usuarios no se reflejan en un informe de back-end.

**Rta:** Una vez finalizado el debug del punto 2, cada vez que un usuario haga una transacción el mismo deberá visualizarse de la forma correcta en el sistema.

1. El ancho de banda necesario para una experiencia de usuario óptima es una velocidad de descarga de 100 MB.

**Rta:** Lo principal es la funcionalidad, el tema del ancho de banda puede resultar algo lento, pero si la funcionalidad no funciona no serviría de mucho darle esa experiencia optima, así que es mejor ir lento, pero seguro, hasta que se resuelvan los primeros puntos.

1. Los foros oficiales de soporte no siempre registran los comentarios del usuario.

**Rta:** Esta podría ser una funcionalidad parametrizable al comienzo, para luego verificar y corroborar cual es el incidente ocurrente.

1. Hay algunos errores ortográficos en la página de inicio.

**Rta:** Tema visual, habría que ver tipo errores existe, mientras no sea funcional va al final.

# Reto 4 - Agile testing

1. Utiliza el mismo entorno que los usuarios finales para realizar las pruebas perfomance.

**Rta:** Lo recomendable es realizar es tipo de pruebas en el ambiente de desarrollo o testig, para no influir en la experiencia del usuario.

1. Las pruebas de humo se ejecutan para cubrir todas las posibles opciones del usuario y se realizan en cada implementación de una nueva versión, todas las pruebas se realizan manualmente para reflejar la misma experiencia del usuario.

**Rta:** No sería del todo necesario aplicarlo en cada versión, ya que como el punto 3 lo indica se realiza una regresión sobre los nuevos módulos a implementar.

1. Las pruebas de regresión solo cubren los módulos que probablemente se vean afectados por los cambios realizados en la última versión.

**Rta:** Es un procedimiento adecuado.

1. Los casos de prueba se crean en Excel para que sean fáciles de editar. Los casos de prueba se eliminan después de la implementación de esa versión, por lo que se crean nuevos completos en cada iteración.

**Rta:** Si bien es practico realizar TestCase en Excel, un correcto procedimiento es guardar las versiones por cada implementación y tenerlos alojados en una carpeta compartida, los casos de prueba también pueden ser subidos a una herramienta de gestión, por ejemplo, Jira.

1. Las pruebas comienzan a estar involucradas en el proceso de desarrollo desde la etapa inicial.

**Rta:** Es un procedimiento adecuado.