### **Ejercicio 1**

### Clasificarlos en funcionales o no funcionales

* Definir criterio de aceptación. En caso de que sea funcional detallar un caso de prueba.
* Identificar si corresponde el tipo de NFR.
* Elegir 4 requerimientos y explicar cómo se convertirían en requeriments creep y gold plating.

### **1: El software debe permitir la creación de perfiles de usuario.**

Tipo: Funcional

Criterio aceptación:

Input : Datos de usuario .

Output: Creación exitosa del usuario.

Caso de prueba:

Tomar datos de una persona, real o ficticia y crearle un usuario.

Este requerimiento podría volverse creep requirements si se piden datos innecesarios para la creación del cliente. Ejemplo datos del padre/familiar del cliente, tipo de sangre, etc.

### **2: El software debe permitir la reserva de citas en el taller.**

Tipo: Funcional

Criterio aceptación:

Input: Seleccionar fecha y hora.

Output: Reserva de cita

Caso de prueba:

Seleccionar una fecha y horario y llevar adelante la efectivización de la reserva.

Este requerimiento podría volverse creep requeriments/gold plating si por ejemplo se pide tener en cuenta a la hora de reservar datos ajenos al objetivo como tal, por ejemplo, si la persona asistirá con alguien más al taller, las condiciones climáticas del día seleccionado.

### **3: El software debe permitir la consulta de información del vehículo.**

Tipo: Funcional

Criterio aceptación:

Input: Consulta de info del vehículo.

Output: Visualización de la información del vehículo.

Caso de prueba: Seleccionar un vehículo y poder ver el detalle de la información del mismo.

### 4: El software debe enviar notificaciones de recordatorios de servicio.

Tipo: Funcional

Criterio aceptación:

Input: dado un evento

output: se envia la notificacion

### 5: El software debe generar informes de rendimiento del taller.

Tipo: Funcional

Criterio aceptación:

Input: dado un evento .

output: se genera un informe con los rendimientos del taller.

### 6: El software debe tener una disponibilidad del 99.9%.

Tipo: No Funcional

Este requisito se puede considerar un gold plating.

### 7: El software debe ser fácil de actualizar sin causar interrupciones en el servicio.

Tipo: NO Funcional

### 8: El software debe cumplir con los estándares de accesibilidad.

NO Funcional

### 9: El software debe tener una interfaz coherente y consistente.

NO Funcional

### 10: El software debe tener un tiempo de carga rápido

Tipo: No funcional

### 11. El software debe contar con un soporte al usuario eficiente.

Tipo: No funcional

### 13: El software debe contar con una arquitectura escalable.

Tipo: No funcional

Este req podría ser gold plating.

### 14: El software debe ser fácil de integrar con otros sistemas

Tipo: No funcional

### 15: El software debe contar con medidas de autenticación de usuarios seguras.

Tipo: No funcional

### 16: El software debe cumplir con las leyes de protección de datos y privacidad.

Tipo: No funcional

### 17: El software debe garantizar la confidencialidad de la información de los usuarios.

Tipo: No funcional

### 18: El software debe ser fácil de desinstalar en caso de ser necesario.

Tipo: No funcional

### 19: El software debe contar con un sistema de respaldo seguro y confiable.

Tipo: No funcional

Dependiendo el sistema de respaldo seguro y confiable puede llegar a ser gold plating.

### 20: El software debe cumplir con las leyes de protección de propiedad intelectual.

Tipo: No funcional