Grupo 7
Ignacio Tachini, Agustin Toya, Felipe Villaronga, Hernan Ciarlo y Benjamín Rielli

EJERCICIO 1

Parte 1

Dada la siguiente lista de requisitos, clasifíquelos en funcionales y no funcionales:

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

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

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

Requerimiento 4: El software debe enviar notificaciones de recordatorios de servicio. **F**

Requerimiento 5: El software debe generar informes de rendimiento del taller. **F**

Requerimiento 6: El software debe tener una disponibilidad del 99.9%. **NF**

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

Requerimiento 8: El software debe cumplir con los estándares de accesibilidad. **NF**

Requerimiento 9: El software debe tener una interfaz coherente y consistente. **NF**

Requerimiento 10: El software debe tener un tiempo de carga rápido. **NF**

Requerimiento 11: El software debe contar con un soporte al usuario eficiente. **NF**

Requerimiento 13: El software debe contar con una arquitectura escalable. **NF**

Requerimiento 14: El software debe ser fácil de integrar con otros sistemas. **NF**

Requerimiento 15: El software debe contar con medidas de autenticación de usuarios seguras. **NF**

Requerimiento 16: El software debe cumplir con las leyes de protección de datos y privacidad. **NF**

Requerimiento 17: El software debe garantizar la confidencialidad de la información de los usuarios. **NF**

Requerimiento 18: El software debe ser fácil de desinstalar en caso de ser necesario. **NF**

Requerimiento 19: El software debe contar con un sistema de respaldo seguro y confiable. **NF**

Requerimiento 20: El software debe cumplir con las leyes de protección de propiedad intelectual. **NF**

Parte 2

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

Nombre del Módulo: Creación de perfiles de usuario

ID de caso de prueba: 1

Nombre del Probador: Juan Pérez

Escenario de prueba: Se debe utilizar el sistema e intentar crear un perfil de usuario.

Descripción del caso de prueba: Compruebe si se crea el usuario, se guarda correctamente en la base de datos y el sistema permite iniciar sesión con dicho usuario.

Pasos de prueba:

- Entrar al sistema
- Clickear en crear un usuario
- Escribir el username
- Escribir 2 veces la contraseña
- Clickear en “crear”
- Ir a la pantalla principal e intentar iniciar sesión con las credenciales del usuario recién creado

Requisito previo: No debe existir un usuario con el mismo username.

Prioridad de prueba: Alta, ya que sin usuarios no tiene sentido tener un sistema.

Datos de prueba: No hay entradas para este caso.

Resultado esperado de la prueba: Se crea un usuario con las credenciales provistas en la pantalla de creación.

Parámetros de prueba: username: usertest, contraseña: usertest

Resultado real: Se crea un usuario.

Información del entorno: se debe utilizar el mismo sistema operativo que se utiliza normalmente en la empresa, así como utilizar una máquina de características similares (misma ram, procesador similar).

Estado: Aprobado, reprobado.

Comentarios: Si el usuario ya existe, el sistema no debería dejar crear un usuario nuevo.

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

Nombre del Módulo: Reservas de citas

ID de caso de prueba: 2

Nombre del Probador: Luis Suarez

Escenario de prueba: Reserva de citas en el taller

Descripción del caso de prueba: Verificar que le software permite a los usuarios reservar citas en el taller de manera correcta.

Pasos de prueba:

- El usuario ingresa al sistema
- El usuario navega hasta la sección reservas de citas en el taller
- El usuario indica fecha y hora deseada para la cita
- El usuario indica tipo de servicio que desea
- El usuario confirma la reserva de la cita

Requisito previo:

- El usuario debe haber ingresado con éxito al sistema
- El taller debe estar disponible para reservas
- No debe existir una reserva para la misma fecha
- El taller debe tener al menos un empleado en el taller para atender la cita

Prioridad de prueba: Alta, ya que para que el taller funcione se deben de poder reservar citas.

Datos de prueba:

Resultado esperado de la prueba:

- El software confirma la reserva de la cita y proporciona un número de referencia o confirmación al usuario.
- El usuario recibe una notificación de confirmación por la reserva, con toda la información de la cita.
- La reserva queda registrada en el calendario del taller y en el perfil del usuario.

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

Nombre del Módulo: Consulta de información del vehículo

ID de caso de prueba: 1

Nombre del Probador: Wilfried Bony

Escenario de prueba: Se debe utilizar el sistema e intentar consultar información sobre un determinado vehículo

Descripción del caso de prueba: Comprobar si en caso de existir dicho vehículo el sistema logra encontrar su información y proveérsela al usuario; sin afectar la base de datos

Pasos de prueba:

- Usuario ingresa al sistema (inicia sesión)
- Usuario navega hasta sección de consultas de vehículos
- Usuario ingresa credencial y matrícula del vehículo
- Usuario confirma la consulta

Requisito previo:

- Usuario debe haber ingresado con éxito al sistema
- Usuario ingreso los parámetros correctamente
- Vehículo debe existir en la base de datos

Prioridad de prueba: Mediana, ya que

Datos de prueba: No hay entradas para este caso.

Resultado esperado de la prueba: El usuario accede a la información del vehículo especificado

Parámetros de prueba: matrícula: tuitontest, credencial: credentialtest

Resultado real: Se retorna la información del vehículo.

Información del entorno: se debe acceder a la base de datos de la cual recuperar información de los vehículos

Estado: el estado de las pruebas, como aprobado, reprobado, NA, etc.

Comentarios:

*Si no existe un vehículo con los datos especificados, notificar al usuario del error.

*La base de datos no se debe ver afectada

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

Nombre del Módulo: Notificaciones de recordatorios de servicio

ID de caso de prueba: 4

Nombre del Probador: Juan Pérez

Escenario de prueba: Se debe comprobar que el sistema envía correctamente las notificaciones por recordatorios de servicio.

Descripción del caso de prueba: Simular el caso en el que a un usuario le falta 1 día para que se renueve automáticamente su suscripción y chequear que le llega una notificación recordando eso.

Pasos de prueba:

- Modificar en la base de datos la fecha de renovación de la suscripción para un usuario de prueba, para que suceda en los minutos próximos
- Esperar que se cumpla el tiempo
- Ingresar con la cuenta del usuario y chequear que en la bandeja de notificaciones haya una recordándole de su renovación de suscripción

Requisito previo: El usuario debe tener una suscripción

Prioridad de prueba: Media, si bien las notificaciones son importantes para el negocio como tal, el negocio puede funcionar bien sin esta funcionalidad

Datos de prueba: En este caso, los datos necesarios serían la información del usuario, su usuario para enviarle notificaciones por la aplicación o un número de celular para poder enviarle notificaciones por mensaje.

Resultado esperado de la prueba: El usuario recibe una notificación.

Parámetros de prueba:

Resultado real:

Información del entorno:

Estado:

REQUISITO 5: El software debe generar informes de rendimiento del taller

Nombre del Módulo: Informes de rendimiento del taller

ID de caso de prueba: **5**

Nombre del Probador: Andrea Pirlo

Escenario de prueba: Se debe comprobar que los informes generados son correctos

Descripción del caso de prueba: Realizar un informe detallado del rendimiento del taller y comprobar si el generado por el sistema es correcto

Pasos de prueba:

- El sistema guarda todos los datos de cada cita del taller
- El sistema analiza todos los datos una vez que es adecuado realizar un informe
- El sistema genera el informe

Requisito previo:

- El sistema debe almacenar a tiempo todos los datos del taller

Prioridad de prueba: Media, es una buena herramienta para que el taller tcrezca y mejore

Datos de prueba:

Resultado esperado de la prueba:

- El software genera de manera correcta los informes de rendimiento, en tiempo y forma.

Comentarios:

En cambio, en el caso de los requerimientos no funcionales, entendimos que son todos los que van del 6 al 20.

- Requerimiento 6: El criterio de aceptación para poder cumplir con este requisito sería que el sistema esté disponible 8751 horas disponible al año, lo cual es difícil de corroborar. Lo mejor sería determinar rangos de operabilidad, considerando que la disponibilidad alcance sus picos cuando la demanda aumenta.

- Requerimiento 7: Para que este requisito se cumpla la infraestructura debería implementar algún tipo de pipeline que permita el deploy en caliente a producción, sumado a esto, también se podrían implementar los deploy en un horario en el que el software tenga poca operativa.

- Requerimiento 8: El software debe ser puesto a prueba por personas con discapacidades, o a su defecto ser probado anotando el comportamiento en una rubrica de evaluación de accesibilidad, obteniendo una calificación alta.

- Requerimiento 9: El software debe seguir estándares y buenas prácticas de UX/UI así como también ser calificado positivamente, en este aspecto, por los usuarios finales.

- Requerimiento 10: El “tiempo de carga rápida” podría considerarse como algo menor a 10 segundos, y esto se puede medir con test de performance, cronometrando el tiempo de demora entre la respuesta de una request y la otra.

- Requerimiento 11: Los usuarios finales deben tener una percepción positiva del servicio de soporte, y que los mismos no se sientan frustrados por demoras o no solución de los problemas.
- Requerimiento 13: El poder tener una estructura escalable se debe medir en que tanto tiempo costaría llevar a cabo el agregar un componente al servidor, agregarle un módulo de RAM, más almacenamiento de disco o un trabajo en nodos o aumentar la cantidad de equipos que le consultarían. Esto se puede medir haciendo los cambios y controlando los tiempos de respuesta que da el programa. Una forma de medir que tan expansible es un sistema, es que, si agregar una componente conlleva muchos pasos y se pasa por varias complicaciones, el sistema no es escalable, en cambio, si agregar un módulo, agregar un node nuevo, agregar más espacio en disco, no llevaía mucho tiempo ni daría muchas complicaciones.

- Requerimiento 14: Debe ser un software modular, para esto debe seguir buenas prácticas en el desarrollo permitiendo su implementación independientemente de los sistemas externos donde se desee integrar.
- Requerimiento 15: El software debe seguir estándares de autenticación, por ejemplo: OAuth.
- Requerimiento 16: El software debe seguir estándares internacionales respecto a la protección y privacidad de datos.
- Requerimiento 17: Se deben utilizar funciones criptográficas y protocolos de encriptación para cerciorar la confidencialidad de la información que circula y se almacena en el software.
- Requerimiento 18: El software debe tener una implementación dentro de la UI que permita la desinstalación sencilla y que sea capaz de ser vista por usuarios no técnicos.

- Requerimiento 19: El que tenga un sistema de respaldo seguro se puede medir en base a que se haga de forma programada, en horario de poca operativa, para no hacer bloqueo de tablas o degradar el tiempo de respuesta promedio del sistema completo. A su vez, que sea confiable implica que se usen herramientas con una tasa de falla mínima, pudiendo ser esta, 1 en 10 intentos, considerando el aviso al administrador correspondiente en ese caso de falla, para que este pueda tomar medidas al respecto y no quede como un evento oculto a ojos de los responsables.

- Requerimiento 20: El software debe apoyarse en estándares y leyes internacionales sobre la protección de los derechos de autor.

Parte 3

- Requerimiento 1: Pertenece al grupo de funcionales, ya que define una funcionalidad a cumplir por el sistema, crear perfiles de usuario.

- Requerimiento 2: Pertenece al grupo de funcionales, ya que define una funcionalidad, poder reservar citas en el taller.

- Requerimiento 3: Pertenece al grupo de funcionales, ya que define una funcionalidad, permitir a un usuario consultar información sobre el vehículo, el cual asumimos que es el mismo que él llevo al taller.

- Requerimiento 4: Pertenece al grupo de requerimientos funcionales, ya que establece que se le deben enviar notificaciones al usuario, estableciendo así el que debe hacer el sistema.

- Requerimiento 5: También pertenece al grupo de los funcionales, puesto que pide que el sistema pueda generar informes de rendimiento sobre el taller.

- Requerimiento 6: Pertenece al grupo de no funcionales, de performance, pues establece restricciones en la disponibilidad del sistema
- Requerimiento 7: Pertenece al grupo de no funcionales, de performance, pues define parámetros para la actualización y la continuación del funcionamiento del sistema.
- Requerimiento 8: Es un requisito no funcional y pertenece a la categoría de Usabilidad ya que establece un criterio de usabilidad para personas con discapacidades.
- Requerimiento 9: Este requerimiento es claramente uno no funcional, ya que habla de aspectos de la interfaz, cayendo por esto en la categoría de Apariencia y sensación.
- Requerimiento 10: Este requerimiento habla de los tiempos de respuesta que debe tener el sistema por lo cual podemos determinar que corresponde a la categoría de rendimiento o performance
- Requerimiento 11: Es un requerimiento no funcional de Mantenimiento, establece que el soporte al cliente debe ser eficiente.
- Requerimiento 13: Este requerimiento se puede asociar a la idea de poder mantener el software a medida que pasa el tiempo, pudiendo incorporar nuevas funcionalidades o

potenciando las existentes con mejores componentes de hardware.

- Requerimiento 14: Requerimiento no funcional de Mantenimiento y soporte, ya que define que el código debe ser fácil de integrar con otros sistemas, lo que requiere un código mantenible y escalable.
- Requerimiento 15: Requerimiento no funcional de Seguridad-Acceso, pues establece restricciones en el ingreso de usuarios respecto a su autenticación y la confiabilidad de estas
- Requerimiento 16: Requerimiento no funcional de Seguridad, en la subcategoría de Privacidad, ya que establece que el sistema debe contar con de protección y privacidad de datos. También se podría considerar dentro de la categoría Legal ya que establece que debe existir una ley.
- Requerimiento 17: Este requerimiento se puede caracterizar como un requerimiento de seguridad ya que debe guardar la información de los usuarios de tal forma que no se filtre al público. No necesariamente está bajo el mandato de una ley. Podría estarlo, y en ese caso se consideraría también como uno legal.
- Requerimiento 18: Requerimiento no funcional de usabilidad, ya que define una facilidad para el usuario que utilizará el sistema, este debe poder desinstalar el software.
- Requerimiento 19: Este requerimiento se puede considerar como uno de mantenimiento/soporte ya que se habla de la seguridad de los datos que el sistema manejará, en cambio también tiene una parte de componente de seguridad, debido a que ante un ataque deben respaldarse los datos, para prever una eliminación de los mismos con fines maliciosos, como la extorsión o borrado de registros que comprometen los intereses del atacante.
- Requerimiento 20: Requerimiento no funcional de tipo Legal ya que establece que el software debe cumplir con determinada ley respecto a la protección de lo derechos de autor.

EJERCICIO 2

Parte 1

Elija 4 requisitos del problema anterior y extiéndalos utilizando los conceptos de requirements creep y gold plating.

Como casos de gold plating tenemos:

- El requerimiento 1 puede expandirse hasta el punto de que el usuario al crear su perfil seleccione el tipo de paleta de colores que quiere para su interfaz del programa. Como tal, no suma mucho, ya que con una versión estandarizada de la vista es más que suficiente para que el usuario pueda usar el sistema cómodamente. El poder personalizarlo, sería un plus cuestionable y que sumaría una carga innecesaria en el equipo de desarrollo

- El requerimiento 4 puede expandirse para que al usuario le permita elegir por donde quiere ser notificado, si por mail, por SMS o por WhatsApp, haciendo que las notificaciones lleguen por el medio seleccionado. Si bien esto puede significar un añadido interesante, se podría solucionar de manera más sencilla e igualmente funcional, definiendo un único medio predeterminado por votación de los usuarios.

Requerimiento 5: El software debe generar informes de rendimiento del taller.
Extendido con **Requirements Creep**

- El software debe recopilar datos sobre calificaciones de clientes o comentarios del foro, cada un tiempo determinado, de forma automática.
- El software debe implementar machine learning a la hora de sugerir medidas de mejora para el taller.
- El software debe evaluar el rendimiento de cada empleado en el taller e incluir en el informe gráficas comparándolos
- El software debe identificar patrones en fallos o en modelos de vehículos que suelen generar más problemas

Requerimiento 18: El software debe ser fácil de desinstalar en caso de ser necesario.
Extendido con **Requirements Creep**

- El software debe tener una versión web que no necesite instalación
- Cuando la aplicación es desinstalada, se debe realizar una encuesta y recopilar datos de por qué se desinstaló.