

A.1. Planteamiento inicial de escenario

Supongan que ambos se encuentran laborando en dos distintas empresas de desarrollo de software, donde una le brinda servicios a la otra. Ambas empresas se encuentran en países distintos y trabajan bajo el paradigma ágil, es decir, aplican metodologías como Scrum o programación extrema. La empresa que llamaremos Acaba de iniciar un proyecto con la empresa B, el cual se trata de una aplicación Web basada en la que la parte del cliente (front-end) será desarrollada en Angular 2, y la parte del servidor (back-end) será desarrollada en Node.js y MongoDB. La manera en la que back-end y front-end se comunican es a través de un servicio tipo REST, por lo que este no presenta una definición de interface como lo hacen los servicios tipo SOAP. Este servicio está siendo desarrollado por la empresa A casi al paralelo que el front-end, el cual está a cargo de la empresa B, por lo tanto, no se tiene documentación sobre los recursos que el servicio expone, ni de los métodos disponibles para cada uno de ellos. Además, la empresa A es quien tiene contacto directo con el cliente, por lo que las historias de usuario las proporciona esta misma empresa, y a quien la empresa B acude para resolver dudas sobre la funcionalidad. Por otro lado, las historias de usuario se están definiendo y refinando a medida que avanza el proyecto, por lo que es común que haya modificaciones en elementos desarrollados con anterioridad.

A.2. Historias de usuario

- **Historia de usuario 1.** Yo como asistente médico quisiera buscar un paciente por nombre o número de seguridad social al momento de otorgarle una cita para el médico al cual estoy asignado.
 - Criterios de aceptación.
 1. Al presionar el botón de buscar debe iniciarse la búsqueda.
 2. Se tiene un solo campo para búsqueda, por lo que se debe identificar automáticamente si se trata de un número de seguridad social o el nombre del paciente.
 3. Sólo se hace búsquedas de los pacientes adscritos a la clínica actual.
 4. En caso de hacer la búsqueda por nombre se deben ingresar por lo menos 2 palabras.
 5. Si la búsqueda es de más de 30 resultados se debe pedir al usuario que refine los criterios de búsqueda.
 6. Al dar doble click a uno de los elementos de los resultados se considera como elegido el paciente.
- **Historia de usuario 2.** Yo como asistente médico deseo registrarle una cita a un paciente para el médico al cual estoy asignado.
 - Criterios de aceptación.
 1. Se debe seleccionar una fecha mayor o igual a la fecha actual. En caso de ser la fecha actual la hora debe ser mayor a la actual.
 2. La fecha debe ser seleccionada de un calendario con la capacidad de poder navegar por días y meses.
 3. En el calendario para escoger fecha se deben marcar de rojo los días en los que ya no hay disponibilidad porque se llenó la agenda y de azul los días en los que no se deben asignar citas, ya sea por inhábil (festivo o fin de semana) o por vacaciones del médico. Las fechas en rojo y las fechas en azul no deben poder seleccionarse.
 4. Al presionar el botón de asignar cita debe verificar que no exista otra cita en la misma fecha y hora. Además, debe asegurarse que no exista una cita asignada en una fecha posterior a la fecha actual, para el mismo paciente. En caso de ello debe sugerirse cambiar la fecha de la cita ya existente. Una vez asegurado lo anterior se debe guardar el registro de la cita.
 5. Todos los campos son requeridos.
- **Historia de usuario 3.** Yo como asistente médico me gustaría poder buscar citas por paciente o por rango de fechas sólo del médico al cual estoy asignado.
 - Criterios de aceptación.
 1. Debe existir una manera de seleccionar la modalidad de búsqueda: por fechas o por paciente.
 2. Cuando se busca por fechas, podrá seleccionarse buscar por una fecha exacta o por rango de fechas. Al tener un rango de fechas, la fecha inicial debe ser menor a la fecha final y ambas fechas deben estar llenas.
 3. La selección de fechas se debe hacer sobre un calendario en donde deben estar marcados de azul los días inhábiles (festivo o fin de semana) o y las vacaciones del médico. Las fechas en azul no deben ser seleccionadas.
 4. Para buscar por paciente, selecciona el paciente tal como lo indica la historia de usuario 1.
 5. Al presionar el botón de buscar se debe mostrar en una tabla los resultados obtenidos. Los campos a mostrar son: fecha, hora y nombre del paciente. En caso de no obtener resultados debe mostrarse una leyenda indicando que no hubo coincidencias. Si son más de 30 resultados deben paginarse los resultados.

- **Historia de usuario 4.** Yo como asistente médico deseo poder cancelar una cita de un paciente adscrito al médico al cual estoy asignado.
 - Criterios de aceptación.
 1. Una vez mostrados los resultados de la búsqueda (historia 4), se selecciona la opción de cancelar. Se debe preguntar en una ventana emergente si se está seguro de cancelar la cita del paciente actual (mostrar nombre del paciente). De ser así, se marca como cancelada la cita y se elimina de los resultados de búsqueda. En caso contrario sólo se cierra la ventana emergente.
 2. No se mostrará la opción de cancelar de citas con fecha menor a la actual.
- **Historia de usuario 5.** Yo como asistente médico deseo cambiar de fecha citas de pacientes adscritos al médico al cual estoy asignado.
 - Criterios de aceptación.
 1. Una vez mostrados los resultados de la búsqueda (historia 4), selecciono la opción de cambiar fecha. Se debe mostrar un calendario para seleccionar la nueva fecha, el cual se debe mostrar como se indica en la historia
 2. También se aplicarán los mismos criterios para dar de alta una nueva cita, tal como se indican en la historia
 2. No se mostrará la opción de cancelar de citas con fecha menor a la actual.

A.3. Guion para cliente.

1. Saluda a tu compañero y dile que eres el que va a desarrollar el front-end, y que según sabes él con quien puedes resolver dudas del back-end y sobre funcionalidad.
2. Espera respuesta
3. Pregunta la IP en donde está publicado el servicio.
4. Espera respuesta
5. Pregunta cómo se debe llamar el encabezado HTTP en donde se envía el API Key
6. Espera respuesta
7. Confirma que si usas el POSTMAN (*etiquetar - sugerida*)
8. Espera respuesta
9. Pregunta como simular que el asistente ésta autenticado si aún no está listo el login (*etiquetar*)
10. Espera respuesta
11. Responde que no, y que tendrías que buscar uno.
12. Espera respuesta
13. Luego que te responda, pídele una lista de los asistentes dados de alta para poder probar (*etiquetar*).
14. Espera respuesta
15. Agradece y pregunta que si el asistente que te pasó ya tiene citas dadas de alta
16. Espera respuesta
17. Ahora exponle tus dudas sobre la historia de usuario 1, cual es el nombre del recurso REST que se debe consultar
18. Espera respuesta
19. Coméntale que en la misma historia 1 dice que se debe identificar automáticamente cuando es un nombre y cuando es un numero de seguridad social, pregúntale como es el formato de ese número (*etiquetar - sugerida*)
20. Espera respuesta
21. Ahora pregunta sobre cómo se llama el recurso REST para la historia 2 (*etiquetar - sugerida*)
22. Espera respuesta
23. Ahora pregunta cómo le harías para saber las fechas en las que ya no hay citas disponibles
24. Espera respuesta
25. Comenta que vas a probar los datos que te proporcionó y luego comenta que si funcionó y que observas las fechas llevan un formato especial. Pregunta si con ese mismo forma hay que enviarlas al servicio (*etiquetar*)
26. Espera respuesta
27. Pregunta sobre la historia 2, si al guardar las validaciones del criterio 4 las aplican en el backend
28. Espera respuesta
29. Ahora de la historia 3, pregunta si el recurso REST a usar es el mismo que el de la historia 2 (*etiquetar - sugerida*)
30. Espera respuesta
31. Comenta que vas a hacer una prueba y luego comenta que el servicio dice que el método GET no está habilitado. Pregunta cómo le harías ahí
32. Espera respuesta
33. Comenta que los datos los mandarías en JSON, pero como sería el formato
34. Espera respuesta
35. Ahora pregunta en que fechas puedes encontrar citas para hacer una prueba (*etiquetar - sugerida*)
36. Espera respuesta
37. Coméntale que los resultados vienen dentro de una lista anidada en el objeto de respuesta, y pregunta para que son los campos extra que vienen en la respuesta
38. Espera respuesta
39. Dile que es todo por el momento y agradece la atención.

A.4. Guion para servidor

1. Responder al saludo y coméntale que a ti te puede preguntar todo lo referente al servicio y las historias de usuario.
2. Espera respuesta
3. Dile que la IP es 45.65.234.9, y que use el puerto 8080 para desarrollo. También comenta que se tienen que enviar todas las peticiones con este api key QWE123RTY345 (etiquetar).
4. Espera respuesta
5. Confirma que si es por el encabezado y que el campo se debe llamar key. Además, pregunta si va a usar algún programa para probar las peticiones
6. Espera respuesta
7. Dile que está bien y que si quiere le puede pasar por correo unas pruebas guardadas de postman.
8. Espera respuesta
9. Coméntale que además del api key tiene que mandar en una cadena encriptada el id del asistente. Comenta también que la encriptación se hace con SHA1 y pregunta si tiene algún componente para hacer esa encriptación (etiquetar)
10. Espera respuesta
11. Coméntale que tienes uno que usaron en proyectos anteriores de Angular 2, y que le mandas una liga para que lo baje (inventa una liga) (etiquetar - sugerida)
12. Espera respuesta
13. Coméntale que hasta ahora solo tienes un asistente (matrícula: 890687567) dado de alta, pero si requiere otro se lo puedes registrar.
14. Espera respuesta
15. Comenta que si tiene algunas citas dadas de alta y que va a tener más porque están haciendo unas pruebas en el backend
16. Espera respuesta
17. Responde que es el recurso de pacientes (etiquetar)
18. Espera respuesta
19. Responde que el número de seguridad social es una secuencia de 9 números únicamente.
20. Espera respuesta
21. Comenta que el recurso de la historia 2 se llama citas. (etiquetar - sugerida)
22. Espera respuesta
23. Coméntale que el recurso que debe usar para saber las fechas en las que ya no hay citas disponibles es citas/diasnodosdisponibles/ y que se le manda por query string la matrícula del asistente (etiquetar - sugerida)
24. Espera respuesta
25. Comenta que el formato de fecha que se recibe en el servicio es yyyy-mm-ddThh:00
26. Espera respuesta
27. Comenta que las validaciones se hacen en el back end. (etiquetar)
28. Espera respuesta
29. Confirma que el recurso de la historia 3 es el mismo que el de la historia 2
30. Espera respuesta
31. Comenta que en ese recurso se decidió usar POST porque se facilita el manejo en el backend. (etiquetar - sugerida)
32. Espera respuesta
33. Dile que el JSON debe tener esta secuencia de campos FechaIni, FechaFin, nombre, nss, todo en string
34. Espera respuesta
35. Comenta que hay citas en todo el mes de febrero de este año.
36. Espera respuesta
37. Sobre los campos extra comenta que son los que manejan el paginado, y que en la consulta se puede agregar el campo página para que solo regrese los elementos de la página especificada.
38. Espera respuesta
39. Dile que cuando guste estarían disponibles.