



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

Creación de chatbot para pruebas automatizadas

INFORME DE ACTIVIDADES PROFESIONALES

Que para obtener el título de

Ingeniera en Computación

P R E S E N T A

Rossana Gabriela Palacios Silva

ASESOR DE INFORME

Ing. Luis Sergio Valencia Castro



Ciudad Universitaria, CD. MX., 2020

Índice

Introducción	2
Capítulo 1. Descripción de la empresa	
1.1 Breve historia de la compañía y estructura organizacional	3
1.2 Descripción del puesto de trabajo	4
1.3 Antecedentes	6
1.4 SCRUM como metodología de trabajo	8
Capítulo 2. Proyecto chatbot de pruebas	
2.1 Origen	12
2.2 Metodología	13
2.3 Diseño	18
2.4 Funcionamiento	20
Capítulo 3. Resultados y conclusiones	
3.1 Resultados	23
3.2 Conclusiones	25
Capítulo 4. Fuentes consultadas	
4.1 Libros	27
4.2 Páginas web	27

Introducción

En la sociedad moderna encontramos numerosos ejemplos que involucran Inteligencia Artificial (IA), desde computadoras que conducen coches, redes sociales que nos muestran contenido basado en nuestros gustos, hasta sistemas que nos apoyan a hacer más eficiente nuestras labores diarias.

Un concepto que va de la mano con el de Inteligencia Artificial es el de chatbot, un software o programa basado en IA. Un chatbot tiene como objetivo simular a una persona real usando un lenguaje natural común. Por ejemplo, un chatbot es usualmente es programado para responder preguntas comunes o brindar alguna información.

Mi primer empleo al terminar la carrera de Ingeniería en Computación fue en una compañía con sede principal en Santa Mónica, California. Esta empresa creó un chatbot que maneja el ciclo de citas para el sector salud de forma automática. Mi primera responsabilidad fue el manejar un proyecto donde pone a prueba la parte de IA de este chatbot.

El presente trabajo se enfoca en describir mis actividades realizadas en la empresa, desde que comencé a laborar ahí en Junio de 2017 a Noviembre de 2018. En él, se explica el proceso de diseño y desarrollo de un chatbot adicional llamado: chatbot de pruebas, el cual actualmente ayuda a la ejecución de pruebas y control de calidad del chatbot principal o producto.

Todas las actividades previamente mencionadas fueron realizadas basándome en los conocimientos adquiridos a lo largo de mis estudios en la Facultad de Ingeniería del 2012 al 2017. De igual manera, también apliqué mi aprendizaje de los cursos de Inteligencia Artificial, Ingeniería de Software, Procesamiento de Lenguaje Natural, Administración de Proyectos de Software, entre otros.

CAPÍTULO 1. Descripción de la empresa

1.1 Breve historia de la compañía y estructura organizacional

La empresa donde laboré, es considerada una startup estadounidense de tecnología basada en Inteligencia Artificial (IA). Fue fundada en el 2016 con la finalidad de desarrollar un software (chatbot) para manejar el ciclo de vida de citas médicas (pediatras, odontólogos, quiroprácticos, entre otros); Cuenta con 2 sedes, su oficina principal se encuentra en Santa Mónica, California y la segunda en Polanco, CDMX.

El producto desarrollado por la empresa es un chatbot que ofrece una solución basada en Inteligencia Artificial para facilitar la comunicación entre clinica-paciente a través de mensajes de texto, de esta manera, el chatbot crea un canal de comunicación en donde pueden confirmar o reagendar su cita de una manera sencilla y rápida. De manera adicional, en caso de que algún paciente no desee confirmar su cita (ni reagendarla), el chatbot, solicita retroalimentación para que el consultorio esté informado sobre las inquietudes de sus pacientes.

En febrero de 2017, la empresa fue seleccionada para entrar al programa de aceleración *StartX* por parte de la Universidad de Stanford. Dicho programa es uno de los más reconocidos a nivel mundial debido a que el 90% de las empresas participantes logran éxito. Desde entonces, la empresa cuenta con mentores y ángeles especializados que guían a la compañía en su crecimiento y organización.

1.2 Descripción del puesto de trabajo

La empresa se divide principalmente en dos áreas:

- Ventas, encargada de las relaciones comerciales, la planeación comercial, seguimiento a clientes y estrategia competitiva.
- Tecnología, encargada del desarrollo, soporte y mejora continua del chatbot creado por la empresa, véase diagrama 1.

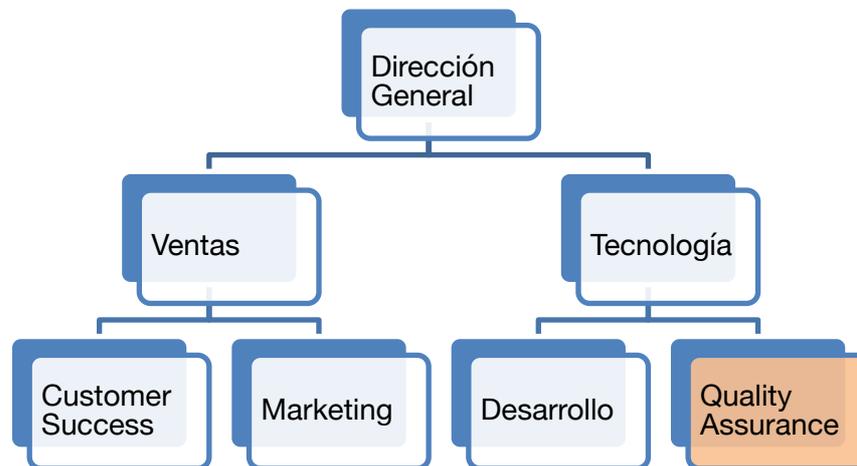


Diagrama 1. Organigrama de la empresa. Fuente: Elaboración del autor

Ingresé a la compañía como responsable de la división de “Quality Assurance”, teniendo bajo mi responsabilidad determinar la estrategia a seguir para realizar pruebas de producto, la metodología, así como seleccionar las herramientas para realizar dichas pruebas. Entre mis actividades figuraron:

- Diseñar casos de uso, pruebas manuales y automatizadas tanto para back-end como para front-end.
- Asegurar la calidad del chatbot para que a medida que aumente su desarrollo en funcionalidad, se mantenga como un producto estable y continúe brindando una ejecución de calidad para el usuario final.

- Monitorear e investigar los casos de error que son reportados por los clientes.
- Ejecutar e implementar pruebas automatizadas para reducir el tiempo de las pruebas antes de cada liberación.
- Asegurar que todas las nuevas preferencias y/o funcionalidades que se le añadan al chatbot funcionen correctamente. Así mismo que ninguna preferencia o proceso anterior se vea dañado por esta mejora.
- Corroborar que todas las funcionalidades nuevas cumplan con los requisitos de los clientes y funcionen adecuadamente.

Para poder realizar estas actividades, tuve que conocer el chatbot a detalle, no sólo lo estudié en el ámbito funcional, también tuve que entender sus procesos internos (*back-end*), la metodología de programación, funciones y todo lo relacionado al proceso de desarrollo.

Durante el proceso de “conocer el chatbot” me encontré con varios aspectos novedosos para mí, algunos de los cuales conocía a nivel teórico pero que en la práctica era la primera vez que los veía implementados: el primero de ellos fue darme cuenta que un chatbot no requiere de un procesamiento de lenguaje robusto para ser eficiente, el segundo aspecto que me llamó la atención fue ver que se pueden ocupar diferentes patrones de diseño en cada módulo del código (de acuerdo al objetivo de cada uno), finalmente, fue entender que por más que desarrolle un diseño interesante y generoso en funcionalidades, siempre existe una curva de adaptación del producto.

Entender el funcionamiento del chatbot, así como su proceso de desarrollo, fue el disparador para concebir el chatbot de pruebas, proyecto enfocado a la realización de pruebas automatizadas basadas en conceptos de Inteligencia Artificial (IA) y sobre el cual se centra el presente reporte.

1.3 Antecedentes

Inteligencia Artificial:

El término “A.I.” fue acuñado por John McCarthy, un informático estadounidense, en 1956 durante la Conferencia de Dartmouth, donde nació la disciplina. Si bien no existe un término preciso para este concepto, cada investigador que ha profundizado en este tema ha creado su definición. Por ejemplo,

“La inteligencia artificial es el estudio de las ideas que permiten ser inteligentes a los ordenadores”

- H. Winston

En términos generales y lo que la mayoría de los autores e investigadores definen es que, la inteligencia artificial es el estudio del comportamiento humano para poder replicarlos en una computadora, de esta manera, una computadora pueda responder de la misma manera que lo haría un ser humano, de manera “inteligente”.

Chatbot:

Un chatbot es una interfaz entre una persona y un servicio.

Usualmente se cree que un chatbot, ofrece o es el servicio como tal, lo cual es una conclusión no acertada. Uno de los aspectos más importantes es que un chatbot tiene delimitado su alcance y objetivo de una manera clara. Un chatbot, también se puede clasificar de acuerdo a su canal de comunicación. (mensajería de texto y otras plataformas como Facebook, Whatsapp, etc.)

Control de Calidad:

Actividades en conjunto que son necesarias durante el proceso de desarrollo para garantizar el buen funcionamiento de un producto. Estas actividades, aseguran que el producto cumpla con las expectativas y calidad necesaria para cumplir con los estándares y/o requerimientos de una tercera parte.

También, se encarga de detectar problemas o un mal funcionamiento de manera temprana.

Pruebas automatizadas:

Pruebas o acciones que permiten simplificar el ciclo de pruebas realizadas a un producto o software. Las pruebas automatizadas no son una herramienta, sino que es una infraestructura y conjunto de pequeños programas diseñados para detectar fallas en cualquier parte del producto o software. Las pruebas deben estar diseñadas para cubrir el mayor número de casos de posibles de funcionalidades del producto.

Caso de uso:

Un caso de uso son pasos o secuencia de acciones que se desarrollan entre un sistema y personas externas a este sistema mejor llamados, actores. Los casos de uso sirven para especificar el comportamiento del sistema como resultado de una interacción con un usuario.

Pruebas de integración:

Tipo de prueba encargada de revisar el funcionamiento de componentes unitarios trabajando en conjunto para formar un software. Una prueba de integración nos garantiza que la comunicación entre elementos individuales e independientes es correcta. Esta prueba siempre se realiza después de que las pruebas unitarias han sido satisfactorias.

Facade:

Patrón de diseño comúnmente usado para tener una interfaz de código simplificada para tratar con otra parte del código más compleja.

1.4 SCRUM como metodología de trabajo

SCRUM en términos generales es un marco de trabajo ágil basado en escenarios, centrado en la colaboración y la funcionalidad cruzada, donde los involucrados son receptivos a la adaptación de mejoras continuas.

Siguiendo el concepto de metodología ágil, los ciclos de entrega utilizando SCRUM son por lo general de 1 mes, a estos periodos cortos de trabajo y entrega se les llama: SPRINT.

El “SCRUM Team”, son grupos pequeños que trabajan hacia un mismo objetivo, al momento de mi llegada, el SCRUM Team se encontraba formado de la siguiente manera:

- PRODUCT OWNER, se asegura de que todo el equipo entiende las mejoras necesarias para el producto, facilita cualquier tipo de información adicional al grupo de desarrollo en caso de ser necesario. En esta posición se encontraba la persona encargada de Customer Success (era la única que estaba en contacto con los clientes y sabía los requerimientos de los mismos).
- SCRUM MASTER, dirige todo el equipo, ayuda al equipo de desarrollo y también lidera la repartición de las nuevas funcionalidades a desarrollar. En la empresa, quien tenía este *rol* era el *Chief Technical Officer* (CTO) de la compañía.
- DEVELOPMENT TEAM, profesionales encargados de realizar los cambios y desarrollar las nuevas funcionalidades requeridas para el producto. A mi llegada a la empresa, únicamente habían dos desarrolladores: el VP (Vice President) de Ingeniería y el CTO.

Flujo de trabajo de un SPRINT. Véase diagrama 2.

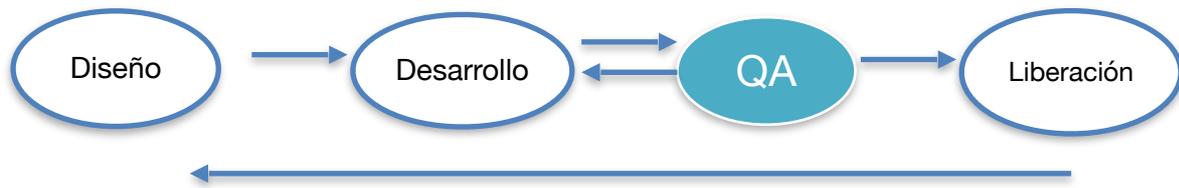


Diagrama 2. Proceso de un SPRINT. Fuente: Elaboración del autor.

El área de *Quality Assurance* (QA) es parte muy importante en el proceso de liberación, tiene una responsabilidad grande ya que cada prueba simula cómo el usuario final va a interactuar con el software. Es importante mencionar que todas las pruebas realizadas son de tipo integral o mejor llamadas *pruebas integrales*, este tipo de pruebas, nos permiten asegurar que cada componente individual funciona de manera correcta estando en conjunto con otros componentes.

Cuando algún código nuevo (ya sea algún cambio simple o nueva funcionalidad) llega al área de QA, es necesario someterlo a pruebas para verificar este cambio, adicionalmente, es necesario realizar otra ronda de pruebas generales para asegurar que nada anterior presente el inicio de alguna falla.

Generalmente, un código nuevo no pasa las pruebas automatizadas en el primer intento, cuando esto sucede, el área de QA es responsable de mandar de regreso el código al área de desarrollo para su corrección o mejora por lo que el flujo de trabajo se ve de la siguiente manera.

Proceso del área de Quality Assurance. Véase diagrama 3.

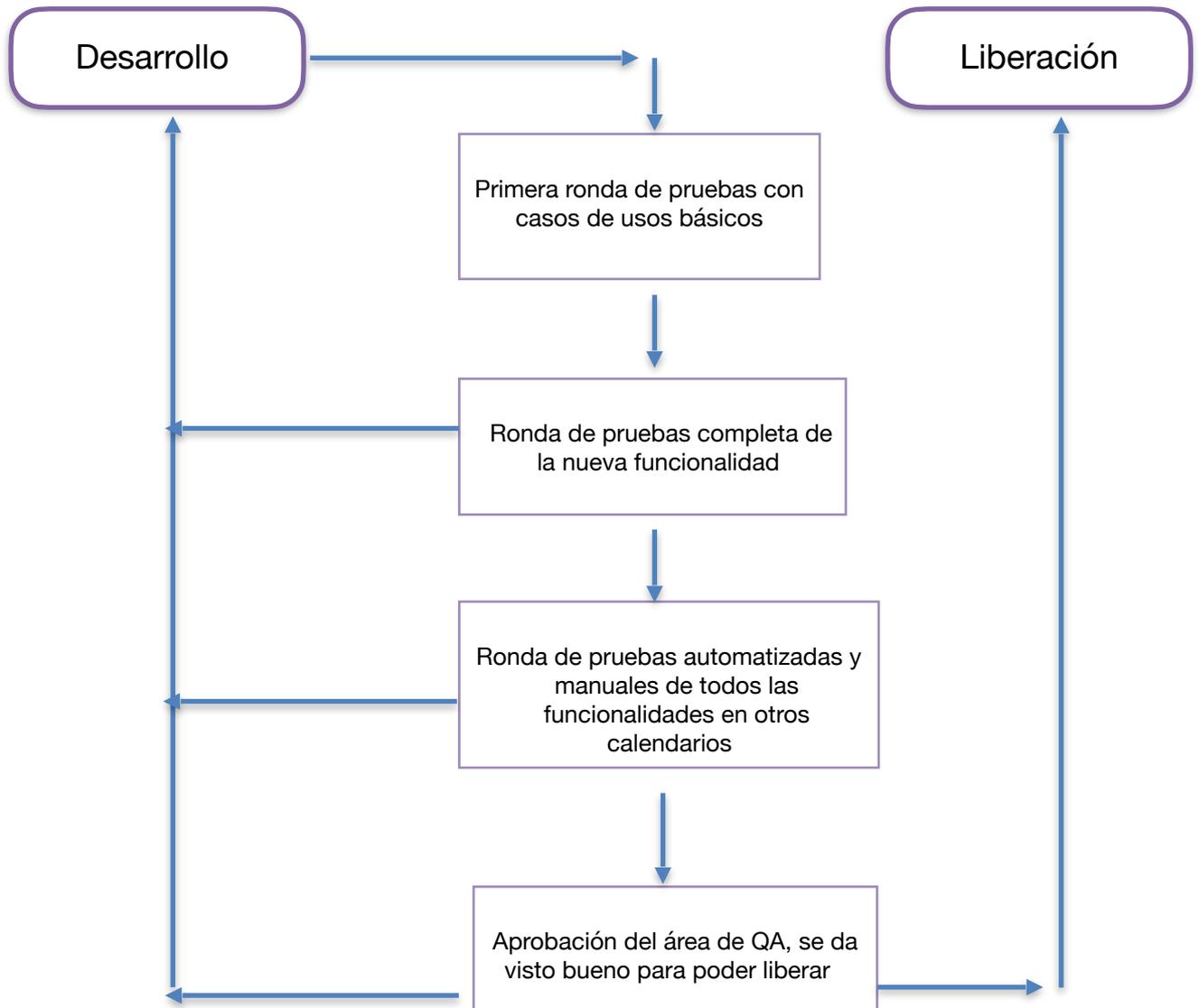


Diagrama 3. Proceso de Testing en el área de Quality Assurance. Fuente: Elaboración del autor.

Tomando en cuenta esta información y dada la importancia de reportar cualquier falla o error, el tiempo comenzó a ser un punto clave. Se comenzó a tener la necesidad de dejar de hacer pruebas manuales y comenzar a poner en código la mayor cantidad de pruebas posibles, de esta manera, reduciríamos el tiempo para aprobar

o rechazar código nuevo; de esta necesidad y problemática nació el chatbot de pruebas.

CAPÍTULO 2. Proyecto: Chatbot de pruebas

El chatbot de pruebas es un proyecto basado y de hecho, es creado con el fin de hacer eficiente el proceso de pruebas de nuevas funcionalidades en mi área de QA.

El chatbot de pruebas, consta de varios scripts y cada uno de ellos es una prueba automatizadas con el fin de ejecutar el mayor número de casos de uso para el chatbot con el fin de reducir la enorme pérdida de tiempo que se tenía; con el chatbot de pruebas se logró reducir hasta el 70% del tiempo en pruebas. También, nos permitiría encontrar problemas y fallas de manera temprana y asegurarnos que el producto cumple con los requerimientos y además está yendo conforme a las necesidades del cliente.

2.1 Origen

El chatbot principal o de producto fue diseñado para el sector médico, se encarga de manejar la comunicación paciente - doctor, con el fin de reducir las pérdidas monetarias por las inasistencias o cancelaciones de última hora. Dentro de su mercado se encuentran los dentistas, gastroenterólogos, podólogos, nutricionistas, psicólogos y otros más.

Para el desarrollo de mejoras del chatbot, era necesario que antes de que se liberaran estas mejoras, el área de Quality Assurance, se asegurara de los siguientes puntos:

1. Las funcionalidades nuevas son probadas de manera exhaustiva, esto es, tener los mayores casos de uso posibles para poder detectar fallas de manera temprana.
2. El tiempo que se toma para probar cada funcionalidad no debe ser mayor al tiempo asignado en el SPRINT (1 semana).

3. Como parte de las pruebas realizadas, una de estas fueron las pruebas de regresión, estas verifican que ninguna nueva funcionalidad afecte o dañe funcionalidades anteriores que ya funcionan correctamente.

Con estos 3 puntos anteriores en mente, pude notar que el tiempo que se requería para cubrirlos era mayor, el área de calidad tenía 1 semana para poder asegurar que todo esté bien, o en su defecto, reportar cualquier bug para que el área de desarrollo pueda corregirlo. Sin embargo, existían tantos casos de usos que 1 semana comenzó a no ser suficiente, por el contrario, se requerían 2 y hasta 3 semanas para poder terminar las pruebas. De esta necesidad de hacer las pruebas más rápidas y eficientes nació el chatbot de pruebas.

El chatbot de pruebas, tiene el objetivo de sustituir la mayor parte posible de pruebas manuales en pruebas automatizadas. De esta manera, atacar los problemas antes mencionados y poder proporcionar reportes de errores oportunos al área de desarrollo.

2.2 Metodología del proyecto

Es importante mencionar y delimitar el campo de trabajo del chatbot de pruebas y desde donde iba a comenzar a funcionar, como antecedente, una prueba manual se hacía sobre un calendario del cual el chatbot de producto ya tenía visibilidad, por lo que hacer esto a nivel código (automatizado), fue necesario añadir un par de pasos adicionales para poder simular esta parte, esto implicaba, incluir en el código la acción de que el chatbot de pruebas busque el calendario *facade* (simulación de un calendario real que te permite editar, crear, eliminar, etc. eventos) y le indique al chatbot principal o producto sobre que calendario *facade* tiene que buscar la nueva cita que se acababa de crear.

Los siguientes puntos, describen una lista de aspectos que se consideraron para desarrollar la implementación del chatbot de pruebas:

1. Delimitar el alcance del chatbot de pruebas, estableciendo el tipo de pruebas que iba a abarcar. La agrupación de pruebas se hizo de la siguiente manera:

- a. **Pruebas por modo de uso** - Enfocadas al manejo que se le da al chatbot desde la perspectiva del paciente, esto considera todas las posibles respuestas, también se consideraron casos no exitosos, por ejemplo, cuando el paciente sólo juega con la plataforma y en realidad no desea ninguna de las opciones que el chatbot ofrece contestando palabras sin sentido o simplemente respuestas fuera del contexto para el que fue creado el chatbot de pruebas, por ejemplo, responder: “Quiero una pizza por favor”.

Esta prueba tiene como objetivo validar que el chatbot es capaz de manejar respuestas que no tiene contempladas o que salen de su margen de entendimiento en la parte de IA y como resultado el estado final de la conversación, por ejemplo: “no entendida”.

Adicionalmente al probar este último caso de uso, esta prueba también tiene contemplado que el status final de la conversación se almacene de manera correcta en la base de datos. Véase imagen 1.

Ejemplo de conversación donde el chatbot no entiende la respuesta pero debería intentar re-dirigir la conversación, el máximo de intentos son 2:

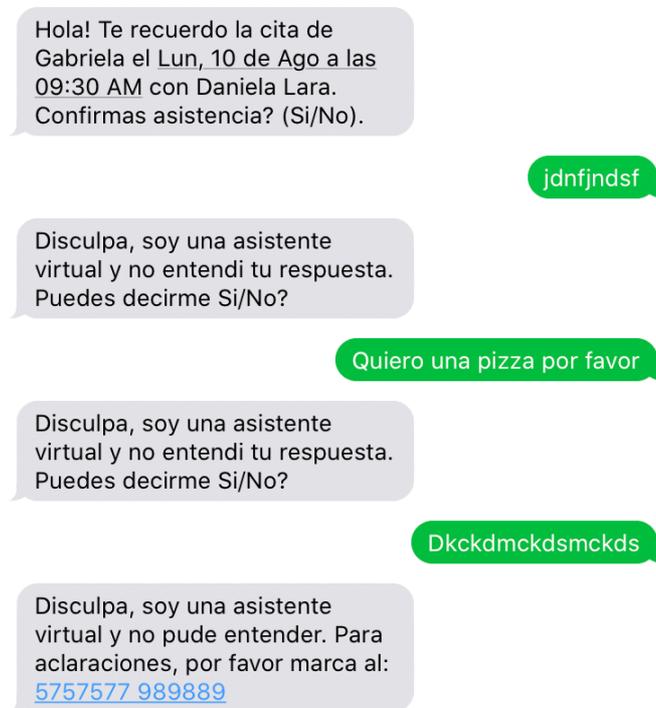


Imagen 1. Simulación de caso de uso erróneo donde las respuestas están fuera del entendimiento del chatbot principal.
Fuente: Elaboración del autor.

b. Pruebas por funcionalidad - Se orientan a cada una de las funcionalidades o preferencias que tiene el chatbot. Estas pruebas validan desde el encendido y apagado de funcionalidades. Actualmente el chatbot consta de 45 preferencias o funcionalidades, todas configurables en cualquier momento, por lo que cada una tiene una prueba específica que valida cada uno de los casos. Esta prueba consiste en:

- a) Encendido y apagado de la funcionalidad: esto valida que los cambios se guarden en la Base de Datos y todas las funcionalidades funcionen adecuadamente.

- b) Activación de múltiples funcionalidades a la vez: Se prueba que aquellas funcionalidades que puedan ser usadas en conjunto con otras, el comportamiento del flujo de la conversación funcione de manera correcta.

Ejemplo: El siguiente caso es un ejemplo de un error que puede ser detectado con el chatbot de pruebas. La funcionalidad “Mensajes de cancelación” se encuentra apagada. Ver Imagen 2.

Por lo que sí una cita es cancelada, no se debería de mandar ningún mensaje para reagendar al paciente. Sin embargo, se puede observar que el chatbot principal sí manda un mensaje, lo cual es incorrecto. Ver imagen 3.

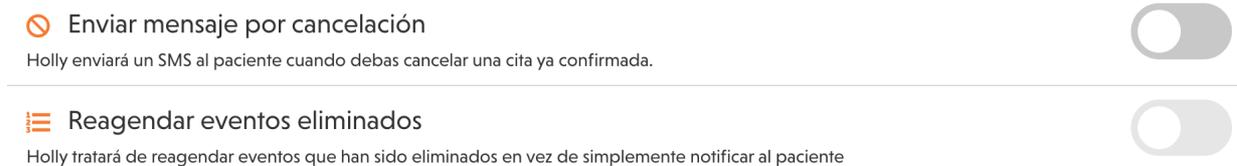


Imagen 2. Funcionalidad “Enviar mensaje por cancelación” apagada.

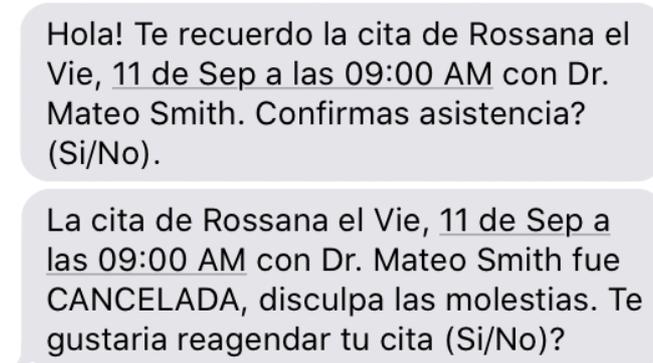


Imagen 3. El segundo texto muestra como el chatbot principal actúa de manera errónea ya que a pesar de tener la funcionalidad apagada, se manda el mensaje por cancelación

- c. **Pruebas de peticiones REST** - Estas pruebas están enfocadas al back-end del chatbot. La arquitectura del chatbot es una API-REST. Todas las llamadas o peticiones le posibilitan hacer los cambios necesarios con información específica, desde añadir eventos nuevos cuando se produce un reagendamiento la nueva cita debe insertarse en el lugar correcto, comprobar que un cupón de descuento fue creado y guardado correctamente en el usuario o que la base de datos fue actualizada correctamente con nueva información de algún usuario.

Es importante mencionar que para los 3 tipos de pruebas anteriores, existen validaciones para saber si el funcionamiento es el indicado. Las validaciones, cuentan con respuestas esperadas y no esperadas, entendiendo que cualquier respuesta no esperada es errónea, en caso de que esto ocurra, el chatbot de prueba lo detecta en ese momento y la prueba es detenida indicando que es lo que está fallando; posteriormente se reporta al área de Desarrollo para su corrección.

2. Una vez comenzada una prueba, debe de hacerse todo el ciclo completo de un caso de uso, por ejemplo, si se está haciendo una prueba automatizada de un reagendamiento, la prueba debe contener todos los casos de uso de un reagendamiento de manera completa. El chatbot de pruebas ejecuta cada uno de los casos programados en la sección anterior *2.2 Metodología del proyecto*.
3. Una parte importante en la metodología de este chatbot de pruebas fue que, el código fuera totalmente independiente al código del producto. Hacerlo de esta manera, nos aseguramos que el código del producto no esta modificado o alterado para que pase la prueba directamente. Sino que realmente se simule la interacción de un paciente.
4. Por último y no menos importante, todas las citas que se crean en este chatbot, deben tener los datos “ficticios” de un paciente, es decir, todas las citas deben de con un nombre y número telefónico ficticio predeterminado (siempre sería el mismo paciente con el fin de identificar las citas de pruebas fácilmente).

5. En cuanto a las herramientas y lenguaje de programación utilizados, se escogieron con base en el lenguaje de programación del chatbot principal, javascript.

2.3 Diseño

El primer punto y más importante en el diseño es que el chatbot de pruebas sea programado en el mismo lenguaje de programación que el chatbot, por lo que el lenguaje sería javascript sobre el framework de Nodejs. El proyecto de este nuevo chatbot, también utilizó frameworks adicionales para pruebas como *Mocha* y *Chai*.

El chatbot de pruebas, simulara un paciente, por lo que todas las citas creadas serán con su nombre, las cuales estarán creadas dentro de un mismo calendario interno creado por la empresa llamado "Memory" (esto con el fin de no saturar las llamadas a nuestros calendarios).

Es necesario que al paciente ficticio que utilizaría el chatbot de pruebas, se le asigne un número del cual enviaría mensajes y así mismo, un proveedor de mensajes y calendario asignados a una variable global para que puedan ser modificados en cualquier momento, asimismo, también probar las pruebas automáticas con otros proveedores de calendarios como Outlook o Acuity, al igual que con otros tipos de proveedores de mensajería como Nexmo o Twilio.

La importancia de hacer esto de la manera antes propuesta, fue con el fin de asegurarnos gradualmente que el chatbot de pruebas estaba funcionando y que las pruebas automatizadas estaban cumpliendo con los puntos requeridos. Por lo que, a la par de que más pruebas automatizadas fueran funcionando, se iban probando cada vez con más proveedores, de esta manera también se comenzó a detectar cuando algo no funcionaba en un proveedor en específico, esto formó parte muy importante del diseño ya que uno de los puntos que era necesario que el chatbot de pruebas abarcara el mayor caso de usos posibles, por lo que el proveedor de calendario o mensajería usado fuera dinámico (se pudiera modificar en cualquier momento), le daba al chatbot de pruebas un punto extra vs las pruebas manuales ya que el chatbot de pruebas era capaz de realizar una prueba completa en cuestión de segundos y a su vez correr la misma prueba con todos los proveedores existentes tanto de mensajería como de calendario; esto no se hacía manualmente por la gran cantidad de tiempo que hubiera requerido hacer la misma prueba con todos los calendarios.

2.4 Funcionamiento

El chatbot de pruebas está construido a partir de pruebas integrales, cada una de estas pruebas corresponde a un archivo o script donde está el código equivalente a una prueba manual. Cada uno de estos scripts, contiene uno o más casos de usos dependiendo la prueba que se esté ejecutando.

Flujo de trabajo del chatbot de pruebas. Véase diagrama 4.

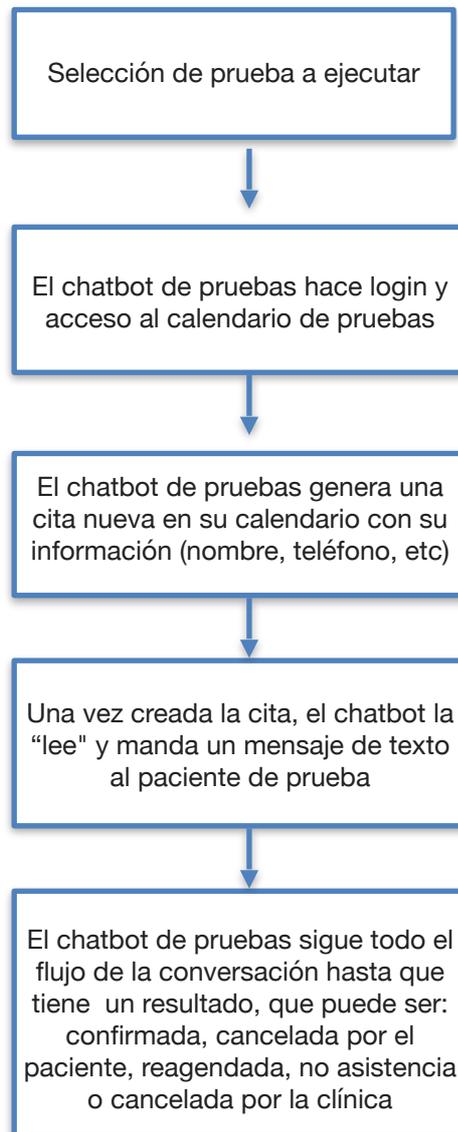


Diagrama 4. Proceso de un SPRINT. Fuente: Elaboración del autor.

Con base en el diagrama anterior:

1. El chatbot de pruebas solicita el ingreso a la página principal de login de la plataforma.
2. Se seleccionan las preferencias de acuerdo a la prueba que se realizará.
3. El chatbot de pruebas le da la información del calendario que se está usando y solicita a el chatbot leerlo y trabajar sobre este mismo.
4. El chatbot de pruebas crea una cita para el mismo calendario automáticamente, lo realiza con el nombre del paciente ficticio y utilizando sus números de teléfono asociados, adicionalmente, manda la información del tipo de prueba que se va a realizar, por ejemplo, si es una prueba de funcionalidad, se prende o apaga dicha funcionalidad.
5. La prueba inicia y se evalúa todo el flujo de conversación, validando todas las respuestas del chatbot, estas validaciones corroboran que el mensaje haya sido recibido y que contenga la información necesaria (nombre del paciente, fecha de cita y zona horaria) correcta.
6. Se evalúa el resultado de la prueba, ya sea que se reporte algún bug (si es así la prueba se interrumpe y lanza un error) o que se corra nuevamente la misma prueba cambiando proveedores (calendarios, mensajes o ambos) verificando que la prueba ha sido exitosa.

Ejemplo: Simulación de conversación entre el chatbot principal o producto y el chatbot de pruebas, véase imagen 4.

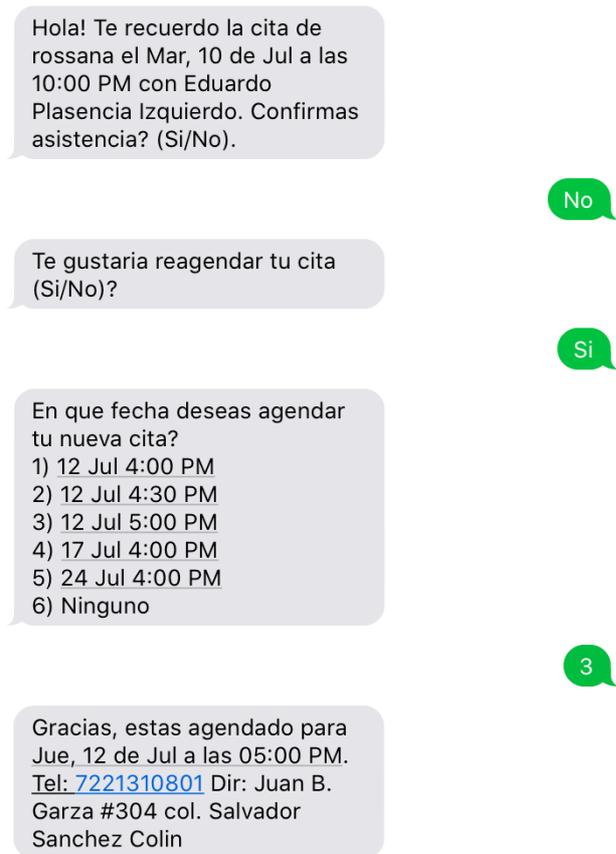


Imagen 4. Simulación de conversación entre el chatbot de pruebas y el chatbot principal. Fuente: Elaboración del autor.

El chatbot de pruebas fue creado y diseñado para simular un paciente, por esta razón, la conversación parece que quien interactúa con el chatbot es una persona “real”. Todas las respuestas del chatbot de pruebas, se encuentran en el script (o archivo) de cada prueba.

CAPÍTULO 3. Resultados y conclusiones

3.1 Resultados

Haber tenido mi primera experiencia laboral en este proyecto fue para mí una gran oportunidad en muchos aspectos, pude estar a cargo de un área importante donde mi base fueron las herramientas que me brindó la Facultad de Ingeniería, también, pude desarrollar mis habilidades interpersonales para poder comunicarme con el equipo de una manera en la que el trabajo fuera realizado con éxito, finalmente, aprendí que generar una relación laboral de apoyo y crecimiento mutuo era indispensable.

La creación del área de QA, fue un proceso lleno de conocimiento, donde al final, puedo rescatar como puntos más importantes los siguientes:

1. Después de analizar todas las pruebas manuales que se realizaban, el objetivo del porcentaje de pruebas automatizadas era de un 85%, la razón de esto es porque existían casos de pruebas específicos donde forzosamente era necesaria la intervención humana, un ejemplo es: verificar que pasado “x” tiempo (1 ó hasta 3 días) una conversación se cerraba exitosamente. Véase gráfico 1.

Resultados de pruebas que pudieron ser automatizadas

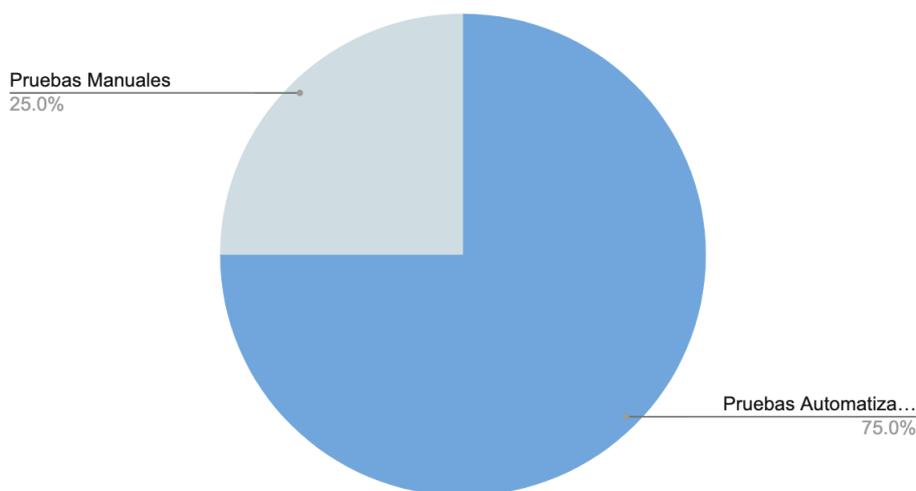


Gráfico 1. Resultado de porcentajes de pruebas automatizadas realizadas al cabo de 9 meses de haber tomado el puesto de Quality Assurance. Fuente: elaboración del autor.

2. El resultado de tener este objetivo fue: de un total de 160 pruebas manuales que se realizaban, se pudieron automatizar 120 pruebas, la automatización llegó a un 75% ya que, por temas de desempeño laboral, fui transferida a otras áreas de la empresa. El tiempo que estuve en esta posición haciendo automatización de pruebas fueron 10 meses.
3. Las pruebas que se lograron automatizar, redujeron el tiempo ejecución en un 90%, por ejemplo, en un inicio, una prueba de confirmación tenía una duración de entre 20 y 30 min; Al tener la prueba automatizada, el tiempo de ejecución se redujo a 1 min.

El chatbot de pruebas se conformaba por el conjunto de todas estas pruebas automatizadas, ayudó a detectar fallas de una forma mucho más eficiente, fallas eran reportadas en formas tempranas y el equipo de desarrollo contaba con mas tiempo para poder corregir y volver a pasar el código al área de QA nuevamente. Véase gráfico 2.

Métricas de tiempo utilizado en pruebas

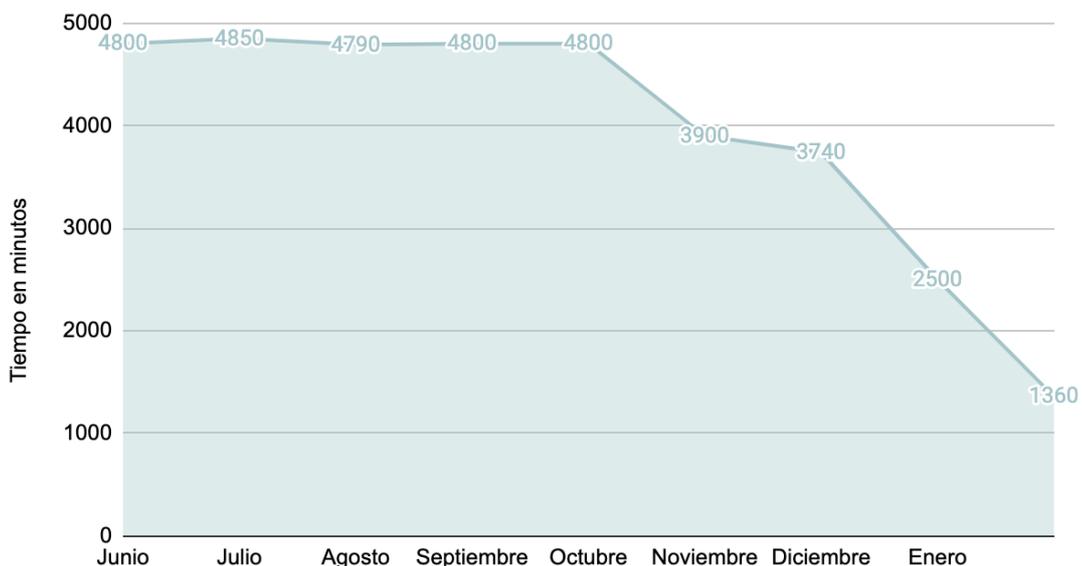


Gráfico 2. Medición de tiempo estimado para el proceso de pruebas. Fuente: elaboración del autor.

3.2. Conclusiones

Durante mi estancia en la empresa, observé y apliqué varios de los conocimientos adquiridos en la Facultad de Ingeniería. Desde mi experiencia, lo más importante fue aprender a optimizar los recursos y entre ellos, el más importante: el tiempo.

También, pude aplicar los conocimientos técnicos propios de la carrera, cómo es la programación orientada a objetos, hablando en términos de código, la cual me ayudó a diseñar de manera correcta el orden de las pruebas y el flujo de la lógica que debía seguir el chatbot de pruebas.

También quiero señalar la importancia del desarrollo de pruebas automatizadas orientadas a la usabilidad de la compañía; en este punto incluyo la parte de desarrollo e implementación de código, así como las estructuras de datos para el análisis de cada una de las pruebas realizadas.

En términos de trabajo en un equipo pequeño, puse ver las múltiples ventajas de utilizar SCRUM como metodología de trabajo y cómo este se convierte en una buena opción si se trata de un proyecto orientado a software y la importancia que tiene el respetar los tiempos asignados y en mi caso particular, respetar el tiempo que existe para probar, todo esto con la finalidad de brindar una respuesta pronta y proactiva para los clientes, pero, sobre todo, brindarles un producto confiable y estable.

Abordando el punto sobre la aplicación y el uso de la Inteligencia Artificial, esta tecnología resultó ser muy conveniente para implementar pruebas automatizadas. Las cuales pude realizar por medio de poner en práctica los conocimientos adquiridos en diferentes temas y proyectos de procesamiento de lenguaje natural, unos de los temas que más utilicé fueron las expresiones regulares, las cuales me ayudaron en la parte de abarcar diseñar pruebas más completas donde también se probó el entendimiento. Estas habilidades se utilizaron al crear conversaciones automatizadas que simulan la interacción con un usuario real. Uno de los avances que logré fue el conteo de la cantidad del máximo de palabras que el chatbot puede entender o en qué

punto de la conversación y bajo qué circunstancias, el chatbot dejó de entender una respuesta.

También pude apreciar una mejora y un mayor entendimiento del diseño de casos de uso implementados en pruebas. Esta parte resultó la más difícil e importante, ya que definir e implementar un buen caso de uso, no sólo se necesitan conocimientos técnicos sino que también una lógica adecuada para que la prueba cubra el mayor alcance posible, es decir que cada caso de uso en la prueba pueda tener sub-casos de uso. Esto se debe a las siguientes razones:

1. Con cada prueba se busca cerciorar que las nuevas funcionalidades trabajen adecuadamente y también que las funcionalidades anteriores no hayan sufrido algún cambio y sigan trabajando bien.
2. Por cada nueva funcionalidad añadida al chatbot principal es necesario hacer pruebas manuales con el mayor número de casos de usos. Estas mismas pruebas fueron sustituidas en un 90% por el chatbot de pruebas. Ya que a diferencia de las pruebas manuales, las pruebas con el chatbot podían ejecutarse en menos de 1 segundo.
3. Es necesario probar la misma funcionalidad en todos los calendarios disponibles hasta el momento. Algunas implementaciones (calendarios) cambian en menor o mayor manera, por ello es necesario asegurarse que la nueva funcionalidad trabaja correctamente en todos.
4. Al momento de la liberación es necesario probar los diversos casos posibles de todas las preferencias. Estas pruebas son necesarias ya que en ocasiones no se modifica el código de la funcionalidad directamente, sino lo que cambia es algún resultado de una variable que se utiliza en la nueva preferencia.
5. Se detectan fallas de manera temprana por lo que la ventana de tiempo para correcciones del lado del equipo de Ingeniería fue mayor.

CAPÍTULO 4. Bibliografía

4.1 Libros

- Stuart Russell & Peter Norvig. (1995). Artificial Intelligence. Upper Saddle River, New Jersey 07458: Prentice Hall.
- Georg E.Shafer. (2013). History of Computer Science. Norderstedt: Deutschen Nationalbibliografie.
- Raul Pino, Alberto Gomez & Nicolas Abajo. (2001). Introducción a la Inteligencia Artificial. Oviedo: Publicaciones de la Universidad de Oviedo
- José Luis Ávila Jiménez. (2015). El ciclo de vida del desarrollo de aplicaciones. España: Elearning

4.2 Páginas web

- <http://www.businessdictionary.com/definition/quality-engineering.html>
(Consultado en Mayo 31, 2020)
- <https://www.getreskilled.com/what-is-a-quality-engineer/>
(Consultado en Mayo 31, 2020)