

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA EN SISTEMAS

**DESARROLLO DE UN ASISTENTE VIRTUAL WEB PARA LA EPN Y UN
ASISTENTE DIRIGIDO POR VOZ EN LOS KIOSCOS DIGITALES DE LA DGIP**

**TRABAJO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN**

KATHERINE LIZETH HURTADO MOÍNA

katherine.hurtado@epn.edu.ec

JONATHAN ISRAEL ZÚÑIGA LOAIZA

jonathan.zuniga@epn.edu.ec

DIRECTOR: PHD. SANG GUUN YOO

sang.yoo@epn.edu.ec

©Escuela Politécnica Nacional 2018

Reservados todos los derechos de reproducción

DECLARACIÓN

Nosotros, Katherine Lizeth Hurtado Moína y Jonathan Israel Zúñiga Loaiza, declaramos bajo juramento que el siguiente trabajo aquí expuesto es de nuestra autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que se ha realizado la investigación necesaria para el desarrollo del proyecto haciendo uso las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración cedemos los derechos de propiedad intelectual correspondientes al respectivo trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Katherine Lizeth Hurtado Moína

Jonathan Israel Zúñiga Loaiza

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por Katherine Lizeth Hurtado Moína y Jonathan Israel Zúñiga Loaiza, bajo mi supervisión.

Ing. Sang Guun Yoo (PhD)
DIRECTOR DEL PROYECTO

AGRADECIMIENTO

Agradezco a Dios y a mis padres que son el pilar de mi vida, a mis queridos amigos que me han acompañado durante este largo camino con su apoyo, al Dr. Yoo por su gran paciencia y a mi amigo y compañero de tesis.

Además, extiendo mi agradecimiento al personal que labora en la Dirección de Gestión de la Información y Procesos de la Escuela Politécnica Nacional quienes han sido un ente indispensable para el desarrollo de este proyecto.

Katherine

AGRADECIMIENTO

Agradezco a mi familia, a mis mejores amigos que siempre me acompañan dándome ánimos para continuar, a mi amiga y compañera de tesis, a mis compañeros de carrera, que son ejemplo de perseverancia y a mis maestros.

Jonathan

DEDICATORIA

Dedico este trabajo de titulación a mi familia, especialmente a mis padres ya que gracias a su apoyo, cariño e infinita paciencia he llegado a culminar una etapa más en mi vida.

Katherine

DEDICATORIA

Dedico este trabajo de titulación a mi familia, por su apoyo durante mi vida académica, pero sobre todo le dedico este trabajo a mi madre y a mi abuela materna las cuales han sido un pilar en mi vida y han sabido ser ejemplo de perseverancia.

Jonathan

RESUMEN

El presente documento se centra en el desarrollo de un asistente virtual (ChatBot) a ser usado por los estudiantes y autoridades de la Escuela Politécnica Nacional (EPN) como ayuda para la atención al usuario brindada por la Dirección de Gestión de la Información y Procesos (DGIP).

Dentro del documento se muestra una arquitectura que integra los servicios web propios del caso de estudio, un ChatBot para los kioscos ubicados en cada facultad, una base de conocimiento, y una plataforma inteligente de procesamiento de lenguaje desarrollada por Google, DialogFlow.

Con esta arquitectura se facilita al ChatBot dar respuestas a través de diálogos personalizados y enfocados en el procesamiento de lenguaje natural.

Los diálogos fueron creados en función de los módulos de consultas, peticiones y preguntas frecuentes entregadas por la DGIP.

Para la interfaz del ChatBot se designaron dos canales (página web de la EPN, y una aplicación móvil destinada a los kioscos ubicados dentro de la institución).

Adicionalmente, se realiza una evaluación al ChatBot a través de parámetros de calidad como son: los niveles de efectividad y grado de satisfacción del usuario.

Palabras clave: ChatBot, procesamiento de lenguaje natural, machine learning, DialogFlow, Scrum.

Abstract

This document focuses on the development of a virtual assistant (ChatBot) to be used by students and authorities of the National Polytechnic School (EPN) as assistance for user assistance provided by the Directorate of Information Management and processes (DGIP).

The document shows an architecture that integrates the web services of the case study, a ChatBot for the kiosks located in each faculty, a knowledge base, and an intelligent language processing platform developed by Google, DialogFlow.

With this architecture, the ChatBot is facilitated to give answers through personalized dialogues focused on natural language processing.

The dialogues were created based on the modules of queries, requests and frequently asked questions delivered by the DGIP.

For the ChatBot interface, two channels were designated (web page of the EPN, and a mobile application for the kiosks located inside the institution).

Additionally, an assessment is made to the ChatBot through quality parameters such as: levels of effectiveness and degree of user satisfaction.

Keywords: Chatbot, natural language processing, machine learning, DialogFlow, Scrum.

Contenido

CAPÍTULO 1	INTRODUCCIÓN	1
1.1	PROBLEMÁTICA	1
1.2	OBJETIVOS	2
1.2.1	<i>Objetivo general</i>	2
1.2.2	<i>Objetivos específicos</i>	2
1.3	ALCANCE	3
CAPÍTULO 2	MARCO TEÓRICO	4
2.1	BASE CONCEPTUAL	4
2.1.1	<i>Agente Conversacional</i>	4
2.1.2	<i>IA aplicada a los agentes conversacionales</i>	11
2.1.3	<i>ChatBots como Plus al negocio</i>	14
2.2	HERRAMIENTA PARA EL DESARROLLO DE CHATBOS	16
2.2.1	<i>Tecnologías para el desarrollo de ChatBots</i>	16
2.3	METODOLOGÍAS DE DESARROLLO DE SOFTWARE	19
2.3.1	<i>Metodologías de desarrollos tradicionales vs ágiles</i>	19
2.3.2	<i>Metodologías Ágiles de desarrollo de software</i>	19
2.3.3	<i>Metodología XP y SCRUM</i>	20
2.3.4	<i>Artefactos de Scrum</i>	21
2.4	TECNOLOGÍAS UTILIZADAS	22
2.4.1	<i>Tecnología y herramienta de desarrollo para la implementación del servidor</i>	22
2.4.2	<i>Tecnología y herramienta de desarrollo para la construcción del SmartBot</i>	24
CAPÍTULO 3	DESARROLLO	28
3.1	DEFINICIÓN DEL PROYECTO	28
3.1.1	<i>Características del proyecto</i>	28
3.1.2	<i>Requerimientos definidos por el equipo de trabajo</i>	29
3.1.3	<i>Selección de la metodología de desarrollo</i>	29
3.2	EVOLUCIÓN DEL PROYECTO	30
3.2.1	<i>Historias de Usuario Épicas</i>	31
3.2.2	<i>Controles</i>	34
3.2.3	<i>Ejecución de los Sprints</i>	34
3.2.4	<i>Estimación de costos de desarrollo</i>	53
3.3	ARQUITECTURA USADA DENTRO DEL PROYECTO	54
3.3.1	<i>Modelo Lógico</i>	55
CAPÍTULO 4	EVALUACIÓN Y RESULTADOS	59
4.1	INSTALACIÓN	59
4.1.1	<i>Instalación del servidor y plugin web</i>	59
4.1.2	<i>Desarrollo e Instalación del aplicativo móvil</i>	64
4.2	DEMOSTRACIÓN DE USO DE LOS APLICATIVOS	69
4.2.1	<i>Aplicativo Web</i>	69
4.2.2	<i>Aplicativo Móvil con tecnología Web</i>	79
4.3	EVALUACIÓN DE RENDIMIENTO	80
4.4	EVALUACIÓN DE ÍNDICES DE CALIDAD Y UTILIDAD	84
4.4.1	<i>Formulación de Encuesta</i>	84
4.4.2	<i>Análisis de Resultados</i>	85
CAPÍTULO 5	CONCLUSIONES Y RECOMENDACIONES	88
5.1	CONCLUSIONES	88
5.2	RECOMENDACIONES	88
5.3	POSIBLES TRABAJOS FUTUROS	89
BIBLIOGRAFÍA		91

ANEXOS.....	95
ANEXO A.....	95
ANEXO B.....	95
ANEXO C.....	95
ANEXO D.....	95
ANEXO E.....	95
ANEXO F.....	95
ANEXO G.....	95
ANEXO H.....	95

ÍNDICE DE ILUSTRACIONES

Ilustración 1. Línea de tiempo asistentes virtuales.	5
Ilustración 2. Línea de tiempo asistentes de voz.	6
Ilustración 3. Características de un asistente conversacional.....	7
Ilustración 4. Uso de los asistentes virtuales.....	8
Ilustración 5. Arquitectura de un ChatBot.....	9
Ilustración 6. Proceso de análisis un ChatBot.....	10
Ilustración 7. Descripción de niveles de PLN.....	11
Ilustración 8. ChatBot KAI – MasterCard.....	14
Ilustración 9. ChatBot GYANT - Salud.....	15
Ilustración 10. Servicio de banco pichincha sábado en la mañana.....	15
Ilustración 11. Servicio Movistar Ecuador domingo en la noche.....	16
Ilustración 12. Arquitectura de NodeJS.....	22
Ilustración 13. Temas WordPress Fuente: https://es.wordpress.org/	24
Ilustración 14. implementación del plugin en WordPress.....	25
Ilustración 15. Cuota del mercado de sistemas operativos móviles.....	26
Ilustración 16. Arquitectura de Android.....	27
Ilustración 17. Prototipo de visualización.....	36
Ilustración 18. Prototipo visualización Chatbot desplegando saludo.....	36
Ilustración 19. Prototipo visualización Chatbot listando menú de ayuda.....	36
Ilustración 20. Burndown Chart Sprint 0.....	37
Ilustración 21. Prototipo visualización de botones de selección de opciones ..	39
Ilustración 22. Prototipo presentación de datos mediante carruseles.....	40
Ilustración 23. Prototipo visualización documento.....	40
Ilustración 24. Prototipo de respuesta a frases inadecuadas para el asistente	40
Ilustración 25. Prototipo de formulario para generación de documento.....	41
Ilustración 26. Prototipo de despedida del ChatBot.....	41
Ilustración 27. Burndown Chart Sprint 1.....	42
Ilustración 28 Prototipo visualización ChatBot móvil.....	44
Ilustración 29 Burndown Chart Sprint 2.....	44
Ilustración 30 Burndown Chart Sprint 3.....	46
Ilustración 31 Prototipo visualización interfaz estadísticas.....	48
Ilustración 32 Burndown Chart Sprint 4.....	48
Ilustración 33 Llenado de datos desde un formulario HTML.....	50
Ilustración 34 Generación del formulario.....	51
Ilustración 35. Generación exitosa del formulario.....	51
Ilustración 36. Descarga del formulario generado.....	51
Ilustración 37. Documento generado.....	52
Ilustración 38. Burndown Chart Sprint 5.....	52
Ilustración 39. Visión simplificada de DialogFlow.....	54
Ilustración 40. Arquitectura de la solución.....	55
Ilustración 41. Estructura de un agente en DialogFlow.....	58
Ilustración 42. Código para el inicio automático del servidor.....	62
Ilustración 43. Codificación Webhook asistente.....	63
Ilustración 44. Codificación generación de formulario.....	63

Ilustración 45. Codificación Webhook reportes.....	63
Ilustración 46. Codificación petición de llenado de formulario	64
Ilustración 47. Construcción interfaz del aplicativo para los kioscos.....	65
Ilustración 48. Configuración de los intents de DialogFlow.....	65
Ilustración 49. Ejemplo de intents carreras de la EPN.....	66
Ilustración 50. Training dentro de DialogFlow.....	66
Ilustración 51. Interfaz del api Chatti	67
Ilustración 52. Transformador de voz a texto en HTML5	67
Ilustración 53. Codificación captadora de voz	68
Ilustración 54. Uso de responsiveVoice.js	68
Ilustración 55. Integración de webview a un app móvil.....	69
Ilustración 56. Ubicación del Chatbot	69
Ilustración 57. Contextos estáticos del Chatbot.....	70
Ilustración 58. Contexto carreras de grado de la EPN	71
Ilustración 59. Contexto carreras de posgrado y tecnologías	71
Ilustración 60. Contexto redes wifi usando correo institucional.....	71
Ilustración 61. Presentación del manual de conexión a internet.....	72
Ilustración 62. contexto conexión a internet sin correo institucional.....	72
Ilustración 63. Herramientas software ofertadas por la EPN	73
Ilustración 64. Licenciamiento de herramientas software	73
Ilustración 65. Ventana de ayuda proceso académico	74
Ilustración 66. Ventana de despedida y manejo de preguntas inapropiadas ...	74
Ilustración 67. Ventana de uso de formulario	75
Ilustración 68. Ventana ayuda de llenado de formulario	76
Ilustración 69. Animación de generación de formulario	76
Ilustración 70. Ventana botón de descarga de formulario.....	77
Ilustración 71. Ejemplo de enlace de documento generado	77
Ilustración 72. Documento generado con datos llenados	78
Ilustración 73. Estadísticas preguntas por contexto	78
Ilustración 74. Estadísticas uso de las plataformas	79
Ilustración 75. Aplicativo móvil para los kioscos	79
Ilustración 76. App móvil adaptativa.....	80
Ilustración 77. Porcentaje de solicitudes usando el Bot.....	81
Ilustración 78. Porcentaje para la generación del formulario	82
Ilustración 79. Tiempos de respuestas para las solicitudes	82
Ilustración 80. Preguntas usadas en la encuesta-sección efectividad	84
Ilustración 81. Preguntas usadas en la encuesta-sección satisfacción del usuario	84
Ilustración 82. Grafica pastel encuesta de edades	85
Ilustración 83. Graficas de columna efectividad	85
Ilustración 84. Graficas de columna satisfacción del usuario.....	86
Ilustración 85. Uso de botlibre	90

ÍNDICE DE TABLAS

Tabla 1. Ejemplos de asistentes virtuales	6
Tabla 2. Comparativa de herramientas	17
Tabla 3. Comparativa metodología tradicional vs ágil	29
Tabla 4 Historia épica HUE1	31
Tabla 5 Historia épica HUE2	31
Tabla 6 Historia épica HUE3	31
Tabla 7 Historia épica HUE4	32
Tabla 8 Product Backlog	32
Tabla 9 Resumen Release Planning	33
Tabla 10 Tabla comparativa de controles.....	34
Tabla 11 Criterios de aceptación Sprint 0.....	35
Tabla 12 Kanban del Sprint 0.....	35
Tabla 13 Retrospectiva usando controles de mejoramiento Sprint 0	37
Tabla 14 Criterios de aceptación Sprint 1.....	38
Tabla 15 Kanban del Sprint 1	39
Tabla 16 Retrospectiva usando controles de mejoramiento Sprint 1	42
Tabla 17 Criterios de aceptación Sprint 2.....	42
Tabla 18 Kanban del Sprint 2 - Prototipo Sprint 2	43
Tabla 19 Retrospectiva usando controles de mejoramiento Sprint 2	45
Tabla 20 Criterios de aceptación Sprint 3.....	45
Tabla 21 Kanban del Sprint 3.....	45
Tabla 22 Retrospectiva usando controles de mejoramiento Sprint 3	46
Tabla 23 Criterios de aceptación Sprint 4.....	47
Tabla 24 Kanban del sprint 4	47
Tabla 25 Retrospectiva usando controles de mejoramiento Sprint 4	49
Tabla 26 Criterios de aceptación Sprint 5.....	49
Tabla 27 Kanban del Sprint 5.....	49
Tabla 28. Retrospectiva usando controles de mejoramiento Sprint 5.	53
Tabla 29. Costos relacionados al talento humano	53
Tabla 30. Estimación de costos no tecnológicos	53
Tabla 31. Configuración de hardware del servidor	59
Tabla 32. Configuración de hardware del cliente web	59
Tabla 33. Configuración de hardware-conectividad.....	59
Tabla 34. Restricciones físicas del proyecto	60
Tabla 35. Restricciones técnicas del sistema	60
Tabla 36. Instalación NodeJS	60
Tabla 37. instalación paquete java	61
Tabla 38. instalación MYSQL.....	61
Tabla 39. configuración de hardware para la plataforma móvil.....	64
Tabla 40. Indicadores de uso del hardware.....	83
Tabla 41. Valores estimados de la encuesta de satisfacción.....	86

Capítulo 1 INTRODUCCIÓN

1.1 Problemática

La evolución de las técnicas de inteligencia artificial ha permitido el incremento en el uso de asistentes virtuales tanto en las plataformas web, como móviles. Esto se puede apreciar en los dispositivos móviles tales como iOS con el sistema Siri y Android con el asistente llamado Google Now. Estos permiten a los usuarios realizar actividades cotidianas como enviar un mensaje de texto o pedir un taxi [1].

La ventaja de utilizar los asistentes virtuales en las diferentes plataformas ha optimizado el tiempo invertido en la atención masiva de clientes dentro del mundo empresarial. Al actuar como si fuera un ente con inteligencia propia simula la atención especializada que un humano podría brindar con la diferencia de que puede atender a varios clientes de forma simultánea [2].

El propósito de los ChatBots es apoyar y escalar los equipos de negocios en sus relaciones con los clientes. Hacer esto ayuda a las empresas a ahorrar mucho dinero, razón por la cual muchos empresarios están adoptando esta tecnología. Y dado el hecho de que estos ChatBots se pueden colocar en lugares como Facebook Messenger, Slack, Telegram, SMS o en su propio sitio web, ofrece la posibilidad de llegar a un público más amplio [3].

Según un artículo publicado por la revista Forbes [4] acerca de la inclusión de los Chatbots en el marketing y la atención al cliente, los beneficios que estos ofrecen son los siguientes:

- Servicio continuo (24/7) de atención al cliente
- Generación de información sobre el cliente
- Participación en conversaciones más personalizadas con el cliente.

En vista de lo anterior se puede resaltar que los ChatBots han ido marcando tendencia a partir de su aparición, grandes empresas como Domino's Pizza, Mercedes-Benz, Comcast, The Wall Street Journal entre otras se han visto en la necesidad de incluir un chatbot en sus plataformas para ayudarles a servir a su gran cantidad de usuarios y alcanzar sus objetivos comerciales [5].

En instituciones educativas como la Escuela Politécnica Nacional (EPN), la falta de atención oportuna a las dudas o inconvenientes genera poca satisfacción en los usuarios. La EPN al ser una institución de educación superior tiene grandes demandas tecnológicas que están en manos de la Dirección de Gestión de la Información y Procesos (DGIP).

Esta entidad está encargada de dar las soluciones pertinentes a los problemas de los estudiantes, docentes y personal administrativo de la institución en temas tecnológicos y relacionados con la institución. En este sentido, al tener un gran número de usuarios, el personal de la DGIP no puede abastecer en su totalidad las demandas internas y externas.

Actualmente, la DGIP utiliza los kioscos ubicados en cada facultad para tratar de solventar las dudas más comunes presentadas por los usuarios, pero esto no ha logrado cubrir la necesidad por completo ya que en la ventanilla de asistencia al usuario aún se tiene un número considerable de usuarios que buscan solucionar sus inquietudes.

En vista de la situación actual, se propone implementar un ChatBot en la página web de la EPN para resolver las inquietudes de los usuarios y, por otro lado, un ChatBot dirigido por voz a manera de kiosco digital para aliviar la carga de demandas que tiene la DGIP y la EPN como institución.

1.2 Objetivos

1.2.1 Objetivo general

Desarrollar un asistente virtual como recurso de apoyo que facilite el manejo de las demandas que tiene la DGIP y la EPN por parte de los usuarios, utilizando tecnologías de procesamiento de lenguaje natural y reconocimiento de voz.

1.2.2 Objetivos específicos

- Desarrollar un plugin que sea compatible con WordPress; este plugin permitirá dibujar la ventana del chat en la página web de la EPN.
- Implementar el reconocimiento de voz para resolver inquietudes de los usuarios.
- Implementar un agente de procesamiento de lenguaje natural y machine learning para resolver las inquietudes en forma de texto.
- Establecer los requerimientos de hardware y configuración de un servidor (webhook) para direccionar las peticiones de los usuarios a las diferentes APIs.
- Establecer conclusiones y recomendaciones en base a los resultados obtenidos en el presente proyecto.

1.3 Alcance

El alcance del proyecto integrador se limita a los objetivos definidos en el punto anterior.

El asistente virtual debe ser capaz de analizar las peticiones recibidas en lenguaje natural para proporcionar respuestas que satisfagan las necesidades del usuario. Para esto se utilizarán tecnologías que faciliten el reconocimiento de voz y el procesamiento de lenguaje natural mediante aprendizaje de máquina.

Se desarrollarán dos tipos de asistentes virtuales, uno con procesamiento de texto en la página web de la EPN y otro con procesamiento de voz en los kioscos de la DGIP ubicados en cada facultad.

Capítulo 2 Marco Teórico

2.1 Base Conceptual

2.1.1 Agente Conversacional

Un agente conversacional o ChatBot es un software que simula la interacción entre un agente de servicio y un usuario para esto hace uso del Lenguaje Natural [6] .

En la actualidad los agentes conversacionales cuentan con interfaces de usuario mediante el uso de TTS (Text To Speech) y STT (Speech To Text) que permiten interactuar mediante el dialogo sin la necesidad del uso de ventanas, iconos o una lista de comandos. Ej. Siri, Cortana, etc. [7]

Gracias a la evolución de la tecnología en los asistentes virtuales y a las bases de conocimiento, estos asistentes son capaces de usar hilos conversacionales con razonamiento basados en casos que simulan diálogos como si se tratara de un agente de servicio humano. [8]

El agente conversacional está dotado de técnicas de lingüística computacional que ayudan con el entendimiento y facilitan las respuestas a las peticiones de los usuarios, todas estas peticiones dependen de un contexto y de un canal de mensajería (emails, chats, etc.). [9]

2.1.1.1 *Estado del Arte de los Asistentes Virtuales*

“Eliza” es un ChatBot considerado el primero, dotado de la capacidad de conversar sobre psicología, fue desarrollado durante los 80s por Joseph Weizenbaum, su objetivo era sustituir a un psicólogo real durante las conversaciones con los pacientes.

Eliza, es la considerada la madre para la mayoría de los agentes actuales, por su similitud en el uso de patrones que simulan la conducta humana.

“Parry”, es otro ChatBot cuyo objetivo, es imitar a un paciente paranoico. Para lo cual este ChatBot tuvo que interactuar con algunos pacientes que sufrían de PDD, esto se realizó para que el Bot pueda interactuar con un médico y logre simular la enfermedad.

En 1995, el Dr. Richard Wallace desarrolla a A.L.I.C.E., usaba SETL, un lenguaje basado en lógica matemática, pero solo llegó a etapas iniciales. Después se desarrolla el “Program A”, esta versión fue desarrollada usando AIML y Java. Después vino el “Program B” en el que ya trabajaron 300 desarrolladores, aquí se sustituyó AIML el cual evolucionó a algo similar al XML, con esto ganó el Loebner Prize en 2000.

Luego vinieron dos versiones de A.L.I.C.E., el “Program C” (desarrollado en C/C++) y el “Program D” (basado en Java2). Con esto para 2001 se creó “The A.L.I.C.E. AI Foundation”.

Otro trabajo destacado es “TinyMud” luego llamado Julia, se trataba de un entorno base para la evaluación de chatbots, este chatbot simulaba ser un mundo virtual donde se discutían distintos asuntos y problemas. La intención de la prueba involucraba no poder distinguir al ChatBot de los usuarios reales.

Existen otros ChatBots relacionados a temas como la música como “Theresa”, “Mimic” copia frases mientras hablas con él y “April” el cual tiene estados de ánimo distintos durante los meses del año.

Casi la totalidad de los ChatBots son de habla inglesa. Pero en la actualidad existen ChatBots para todos los idiomas y algunos hablan varios como “Eloisa”. Por ejemplo, para el idioma español existe Dr. Abuse considerado el mejor ChatBots de habla española en la actualidad.

Dr. Abuse tiene la personalidad de un psiquiatra con la capacidad de seguir casi cualquier conversación. Inclusive ocupa el humor por lo que sus autores ya han delimitado su personalidad.

Zebal también cuenta con sintetización de voz en español, también posee como característica la capacidad de interactuar mediante un protocolo de debate (IRC), cuenta con 40000 palabras accesadas remotamente y puede recoger más datos de la web, esto hace de Zebal un asistente que simula interacción humana con los usuarios, pero sus hilos o flujos conversacionales no son tan buenos.

Por último, la mayoría de los ChatBots creados no especifican sus temas, pero existen ChatBots especializados en temas de gobierno, medicina, cocina, etc.

En la Tabla 1 se muestra algunos ejemplos de ChatBots con su idioma y utilidad. [10]

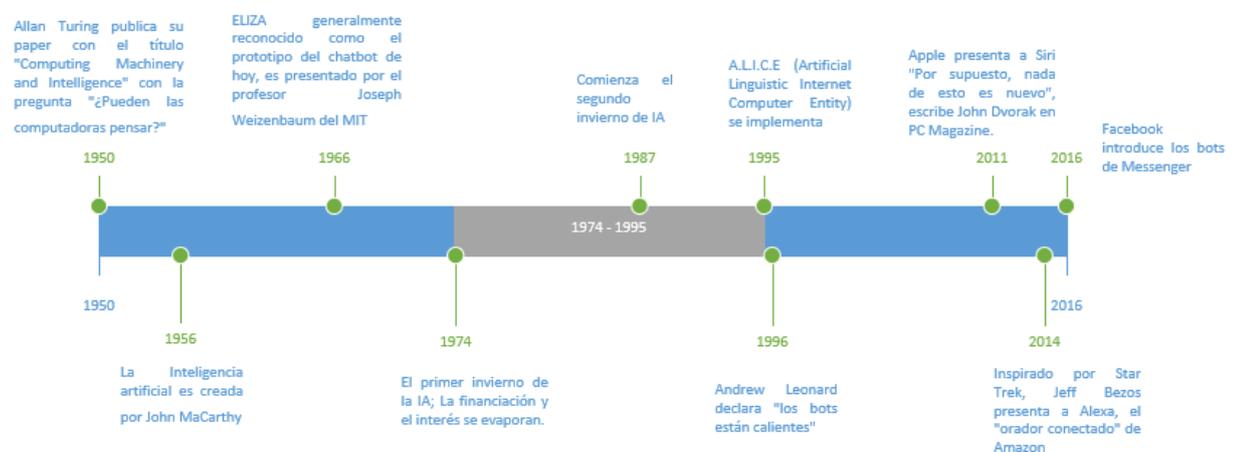


Ilustración 1. Línea de tiempo asistentes virtuales.

Fuente [11]

Algunos ejemplos:

Tabla 1. Ejemplos de asistentes virtuales

Nombre	Tema	Idioma	Url
Eliza	Psicología	Inglés	http://www.manifestation.com/neurotoys/eliza.php3
Zebal	General	Español	http://espasoft.net/fichas/zebal.shtml
Julia	General	Inglés	http://lazytd.com/lti/julia/
Erin	Rock & Roll	Inglés	http://www.extempo.com/webbar/index.html
Dr. Abuse	Psicología	Español	https://www.psicoadiva.com/doctor-abuse/

Evolución de los asistentes de voz

Los asistentes de voz se han vuelto parte importante de los dispositivos móviles debido a su utilidad por lo que han tenido gran acogida, gracias a esto, en estos últimos tiempos se ha visto un notable crecimiento, según el portal “Statista” [12] se provee que este año 4.57 billones de usuarios dispondrá de un teléfono móvil.

En todos los campos tecnológicos se están usando asistentes de voz, esto no es otra cosa que la llamada IA. Y ya no solo se habla de software sino también están incluido dentro los dispositivos como son altavoces, lámparas, cines en casa, etc. Muchas empresas están en la cabeza del desarrollo de estas tecnologías como ejemplo tenemos a Google, Apple, Amazon entre otras. El principal objetivo de estas tecnologías es la facilitación de operaciones e integración con las tareas diarias.

Al igual que el desarrollo de los ChatBots, los asistentes de voz cuentan con la siguiente historia:

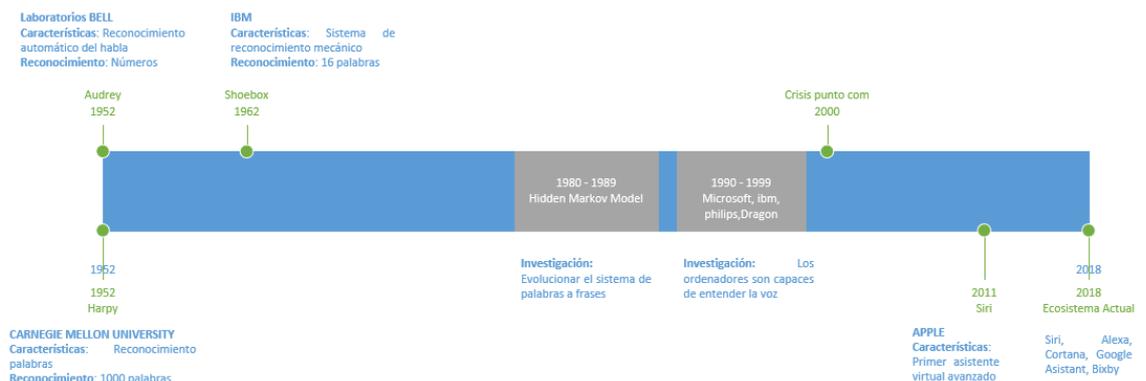


Ilustración 2. Línea de tiempo asistentes de voz.

Fuente [13]

2.1.1.2 Características de un agente conversacional

Un agente conversacional tiene características que le permiten realizar múltiples funciones, entre las más útiles tenemos las siguientes:



Ilustración 3. Características de un asistente conversacional

Fuente: [14] [9]

Racionalidad: el asistente brindara una respuesta dependiendo de su entorno y como este lo perciba [14].

Proactividad: el asistente debe ser capaz de tomar la iniciativa en una conversación y enfocar el tema de conversación [9].

Accesibilidad: esta característica está relacionada con la capacidad que tiene un asistente de brindar ayuda al usuario ya sea por el uso de herramientas de comunicación ej. Gestos, gráficos, sonidos etc. [9].

Adaptabilidad: el asistente debe ser capaz de aprender durante su proceso de uso y de adaptarse a las necesidades del usuario [9].

Expresividad: esta característica hace uso de los elementos como son gráficos, videos y sonidos presente en los distintos canales, los elementos multimedia apoyan al dialogo para que este aumente su contexto [14].

Identidad: un asistente cuenta con la personalidad que el programador le haya brindado durante su creación, pero la personalidad de un agente conversacional con capacidad de aprendizaje dependerá del entorno y los diálogos que se usen con él. [9].

Muchas de estas características pueden cambiar dependiendo del objetivo del negocio.

2.1.1.3 Aplicaciones

Dentro de las áreas de aplicación de los asistentes virtuales están: Industrial, Comercial, Educación, Medicina, etc.

El uso de agentes en aplicaciones comerciales dentro de Internet como en redes corporativas está en auge, se puede distinguir las siguientes:

Gestión de información: Dentro de este campo el agente debe discernir entre correos, noticias o información que se encuentre en la red. El agente también puede ayudar con el almacenamiento, aprendizaje, y manejo de las preferencias del usuario. Entra también el concepto de ayudante o asistente personal.

E-Commerce: Dentro de los entornos virtuales dedicados a los negocios, tenemos muchas aplicaciones para los agentes que van desde la compra, ventas o búsquedas de algún producto, verificación de precios, etc.

Monitorización: El agente es el encargado de avisar al usuario en caso de algún cambio en la variante de estudio. Como ejemplo tenemos cambios en la Bolsa o variante en el clima, al igual que el estado de un sistema.

Mediador entre distintas fuentes de información: El trabajo actual está enfocado en que los agentes logren diferenciar entre los contextos de datos y el entorno en el cual se desarrollan.

Dentro del área médica, se ha visto la necesidad de disponer de agentes que realicen de forma autónoma ciertas tareas dentro de un hospital, uno de los ejemplos cotidianos es la monitorización de pacientes en cuidados intensivos.

El uso de agentes conversacionales en el entretenimiento digital como son juegos de video ha permitido la creación de diálogos más sofisticados en los personajes y múltiples interacciones con el jugador.

A continuación, se muestra un gráfico resumen del uso de los asistentes virtuales en los diferentes campos de la industria.

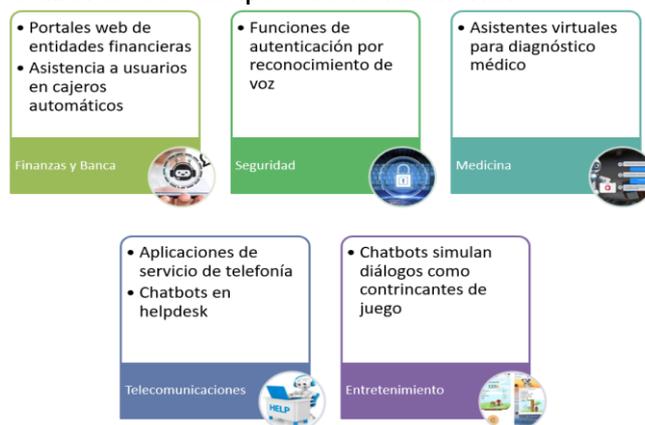


Ilustración 4. Uso de los asistentes virtuales

2.1.1.4 Arquitectura

La ilustración 5, muestra la arquitectura de un ChatBot.

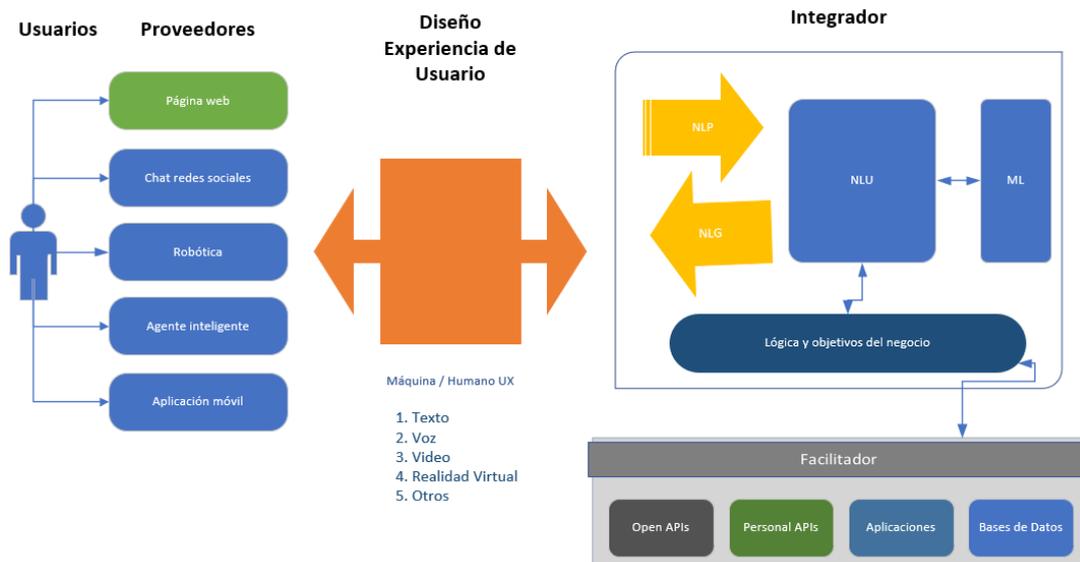


Ilustración 5. Arquitectura de un ChatBot

Fuente:[15]

A continuación, se describen los componentes principales de la arquitectura.

1. **Interfaz de Usuario:** Es el medio por el cual el usuario envía las entradas de información hacia el ChatBot.
2. **Motor de inferencia:** Es encargado de analizar la información, establecer objetivos y obtener una respuesta de acuerdo con la base de conocimiento, este también se encarga de enviar de vuelta el resultado a la interfaz de usuario. Un motor de inferencia utiliza dos elementos: la data (hechos o evidencias) y el conocimiento (conocimiento como el conjunto de reglas almacenadas en una base), gracias a estos dos elementos se puede obtener nuevas conclusiones o hechos.
3. **La Base de conocimiento:** contiene todo el conocimiento del experto humano, en función de plantillas, patrones y reglas.

Dentro de la arquitectura descrita se manejan varios procesos para poder descomponer las oraciones, detectar posibles errores entre otros, la Ilustración 5 detalla este proceso.

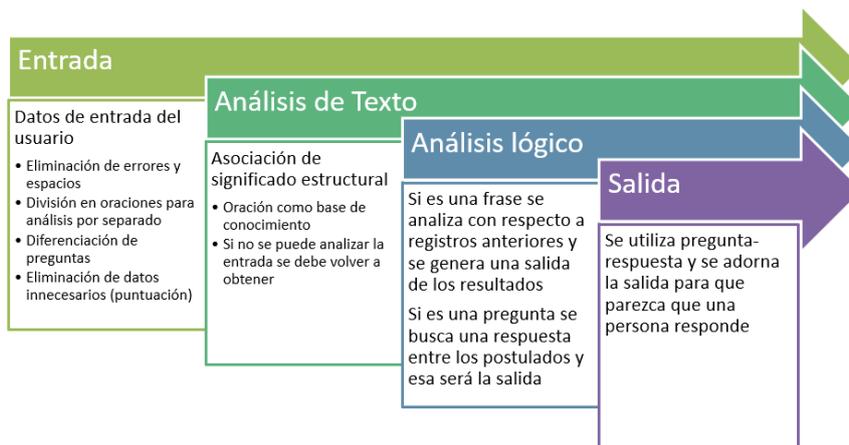


Ilustración 6. Proceso de análisis un ChatBot

Fuente: [9]

2.1.1.5 Agentes Animados

Los ECA's (Embodied Conversational Agents por sus siglas en inglés), en español Agentes Conversacionales Corpóreos o Agentes Conversacionales Animados son interfaces gráficas con la capacidad de utilizar la comunicación verbal y no verbal para interactuar con usuarios.

Estos asistentes en ciertas ocasiones son simplemente un rostro parlante con animación, estos pueden mostrar expresiones faciales simples, también suelen usar síntesis de voz con sincronización labial, existen también sofisticadas representaciones en 3D. [16]

Los agentes animados cuentan con las siguientes habilidades:

- Reconocer y responder a entradas verbales y no verbales por parte del usuario.
- Generar salidas verbales y no verbales.
- Realizar funciones conversacionales (por ejemplo, turno de la palabra, retroalimentación, etc)
- Dar señales que indiquen el estado de la conversación, así como contribuir con nuevas proposiciones. [16]

Después de realizar una serie de estudios e investigaciones sobre el paradigma de "computadoras como actores sociales", se ha llegado a la conclusión de que es posible manipular el comportamiento del usuario con el uso de ciertas palabras y expresiones, también se enfatizó en el uso del halago para aumentar la preferencia del usuario.

También el uso del humor dentro de los comentarios de los chatbots permite que estos sean clasificados como amables, competentes y cooperativos, además en la mente del usuario se crea afectividad lo cual en el caso de las ventas de productos es un punto fundamental. [17]

2.1.2 IA aplicada a los agentes conversacionales

2.1.2.1 *Procesamiento de Lenguaje Natural*

Se entiende como Procesamiento del lenguaje natural (PLN) a la capacidad que tiene un software para procesar información que puede ser voz o texto [18].

El PLN depende del entendimiento de las oraciones ingresadas por los usuarios, para esto el usuario hace uso del lenguaje natural (LN) [19].

2.1.2.2 *Arquitectura de un sistema PLN*

La arquitectura está sustentada en la definición del Lenguaje Natural y puede ser clasificada por niveles [19]:

- **Nivel Fonológico:** análisis de la relación entre el sonido y las palabras
- **Nivel Morfológico:** usa las unidades más pequeñas para la construcción de las palabras, haciendo uso de sus significados
- **Nivel Sintáctico:** encargado de la formación de las oraciones, analizando que palabras pueden ir juntas.
- **Nivel Semántico:** encargado de dar el significado a las oraciones y las palabras que la componen.
- **Nivel Pragmático:** Define la situación y con esto el tipo de oraciones que puede utilizar.



Ilustración 7. Descripción de niveles de PLN

- **Expresión en LN:** es la entrada y lo que el usuario quiere comunicar
- **Análisis morfosintáctico:** conformado por un analizador lexicográfico llamado también scanner y un parser encargado de ver el orden gramatical
- **Análisis semántico:** en este paso se otorga un booleano a cada oración y se obtiene su significado

- **Forma lógica:** aquí están las oraciones ordenadas
- **Análisis pragmático:** en este paso el sistema toma todas las oraciones analizadas y pasa al siguiente paso
- **Expresión final:** conjunto de las oraciones analizadas anteriormente
- **Ejecución:** encargada de buscar la presentación de los resultados
- **Resultado** [19].

2.1.2.3 *Aplicaciones PLN*

Los sistemas PLN son aplicados para diferentes campos en la actualidad, junto con el desarrollo de IA se empezó a crear miles de aplicaciones con altos niveles cognitivos, entre las aplicaciones desarrolladas están [19]:

- Asesores Virtuales.
- Respuestas automáticas.
- IA en compañeros de videojuegos.
- Agentes Conversacionales (Chatbots).

2.1.2.4 *Machine Learning*

Machine Learning es una aplicación de IA que permite que las computadoras aprendan automáticamente sin intervención humana o asistencia predefinida explícitamente.

El aprendizaje automático se centra en el desarrollo de programas informáticos que pueden acceder a datos y usarlos para aprender por sí mismos.

Este proceso de aprendizaje comienza con observaciones o datos, como ejemplos, experiencia directa o instrucción, con el fin de buscar patrones y tomar mejores decisiones en el futuro.

Los algoritmos de aprendizaje automático a menudo se clasifican como supervisados o no supervisados, pero existen mezclas. [20]

Los algoritmos supervisados pueden aplicar lo que se aprendió en el pasado con nuevos datos usando ejemplos etiquetados para predecir eventos futuros.

A partir del análisis de un conjunto de datos de entrenamiento conocido, el algoritmo de aprendizaje produce una función inferida para hacer predicciones sobre los valores de salida.

El algoritmo de aprendizaje también puede comparar su salida con la salida correcta y prevista, y encontrar errores para modificar el modelo

. Algunos modelos fundamentados en este tipo de aprendizaje son:

- redes neuronales artificiales
- arboles de decisión
- etc. [20].

Para el **aprendizaje automático no supervisados** son utilizados cuando la información de entrenamiento no está ni clasificada ni etiquetada. [20]

El **aprendizaje no supervisado** estudia cómo los sistemas pueden inferir una función para describir una estructura oculta a partir de datos no etiquetados. El sistema no encuentra el resultado correcto, pero explora los datos y puede sacar inferencias de los conjuntos de datos para describir las estructuras ocultas de los datos no etiquetados. [20]

Algunos modelos fundamentados en este tipo de aprendizaje son:

- K-means, clustering
- Cadena de markov
- Análisis de sentimientos.
- Reconocimiento de Voz [20]

Los algoritmos de aprendizaje automático semi - supervisados estos utilizan datos etiquetados y no etiquetados para realizar el entrenamiento, generalmente se selecciona una cantidad pequeña de datos etiquetados y una gran cantidad de datos sin etiquetar. Los sistemas que usan este método pueden mejorar considerablemente la precisión de aprendizaje. Por lo general, el aprendizaje semi - supervisado se elige cuando los datos etiquetados adquiridos requieren recursos calificados y relevantes para poder aprender de ellos. De lo contrario, la adquisición de datos no etiquetados generalmente no requiere recursos adicionales. [21]

Los algoritmos de aprendizaje de máquina de refuerzo Se basan en un método de aprendizaje que interactúa con su entorno al producir acciones y descubre errores o recompensas. La búsqueda de prueba y error y la recompensa retrasada son las características más relevantes del aprendizaje de refuerzo. Este método permite que las máquinas y los agentes de software determinen automáticamente el comportamiento ideal dentro de un contexto específico para maximizar su rendimiento. Se requiere de retroalimentación simple para que el agente sepa qué acción es mejor; esto se conoce como la señal de refuerzo. [21]

El aprendizaje automático permite el análisis de cantidades masivas de datos. Aunque generalmente brinda resultados más rápidos y precisos para identificar oportunidades rentables o riesgos peligrosos, también puede requerir tiempo y recursos adicionales para el entrenamiento adecuado. La combinación del aprendizaje automático con AI y las tecnologías cognitivas puede hacer que sea aún más eficaz en el procesamiento de grandes volúmenes de información. [21]

2.1.3 Chatbots como Plus al negocio

Un ChatBot básicamente es un software con inteligencia artificial que fue creado con el propósito de entablar una conversación con el usuario, pero a diferencia de un ChatBot convencional este tendrá que entregar información específica solicitada por el visitante, simulando ser un agente de servicio o soporte. Toda esta información dependerá del servicio ofrecido o la naturaleza del negocio.

Como casos de éxito se tiene:

Una encuesta reciente de 3.000 estadounidenses realizada por MasterCard y Mercator revela que la nueva ola de asistentes virtuales ya ha comenzado a cambiar la forma en que participamos e interactuamos en nuestras vidas. [22]

Dos tercios de los encuestados ya están usando asistentes de voz o Chatbots basados en mensajería / texto, y uno de cada cinco los está usando con fines comerciales. Dentro del artículo “THE CONVERSATIONAL BUSINESS” de Susan Eltinger, se menciona como MasterCard, crea su propio ChatBot llamado “MasterCard KAI”, el cual fue un BUM por la facilidad de compras con diversos locales.

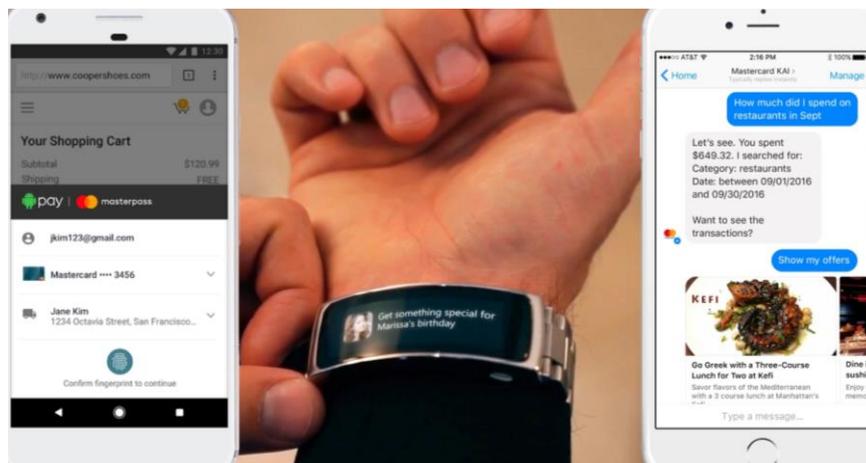


Ilustración 8. ChatBot KAI – MasterCard.

Fuente: [9]

GYANT, es un simpático asistente virtual de salud existente en Facebook, este ayuda con la identificación de síntomas que puedas tener. Este hace uso de una serie de preguntas como edad, donde vives y los síntomas que tienes, para asesorarte de manera gratuita, pero su objetivo es únicamente informativo.

Este asistente virtual de la salud trata de establecer una conversación bastante agradable, acompañado del uso de imágenes e iconos que hacen la charla muy intuitiva, y está disponible 24/7.

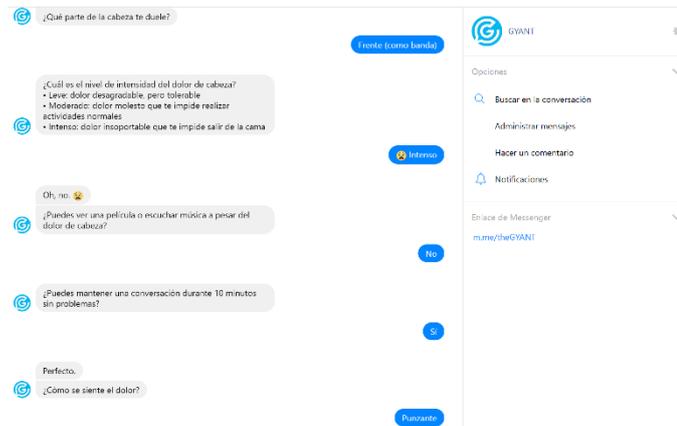


Ilustración 9. ChatBot GYANT - Salud

En el caso específico de Ecuador, los asesores virtuales están tomando mucho auge. Aunque algunos medios todavía consideran el uso de un agente humano a través de un chat como asistente virtual.

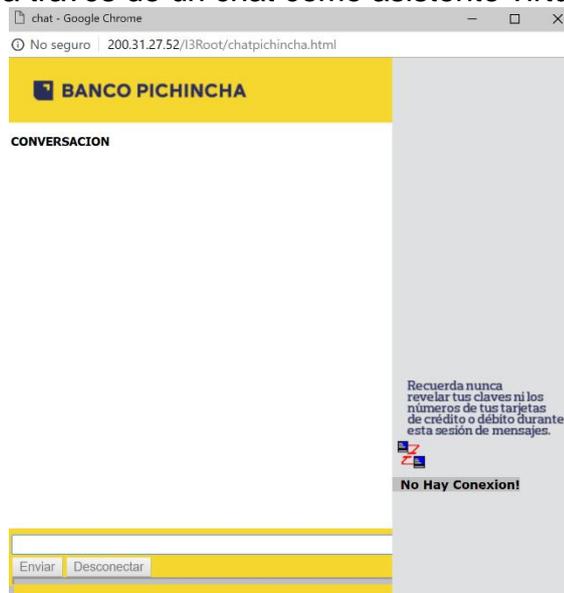


Ilustración 10. Servicio de banco pichincha sábado en la mañana

Esto genera que la disponibilidad del servicio durante periodos del día no sea la adecuada. Y genera descontento entre los usuarios.

Se tiene un ejemplo con la empresa Telefónica – Movistar, quienes se han visto en la necesidad de implementar asistentes virtuales en sus diferentes plataformas como se muestra en la figura.

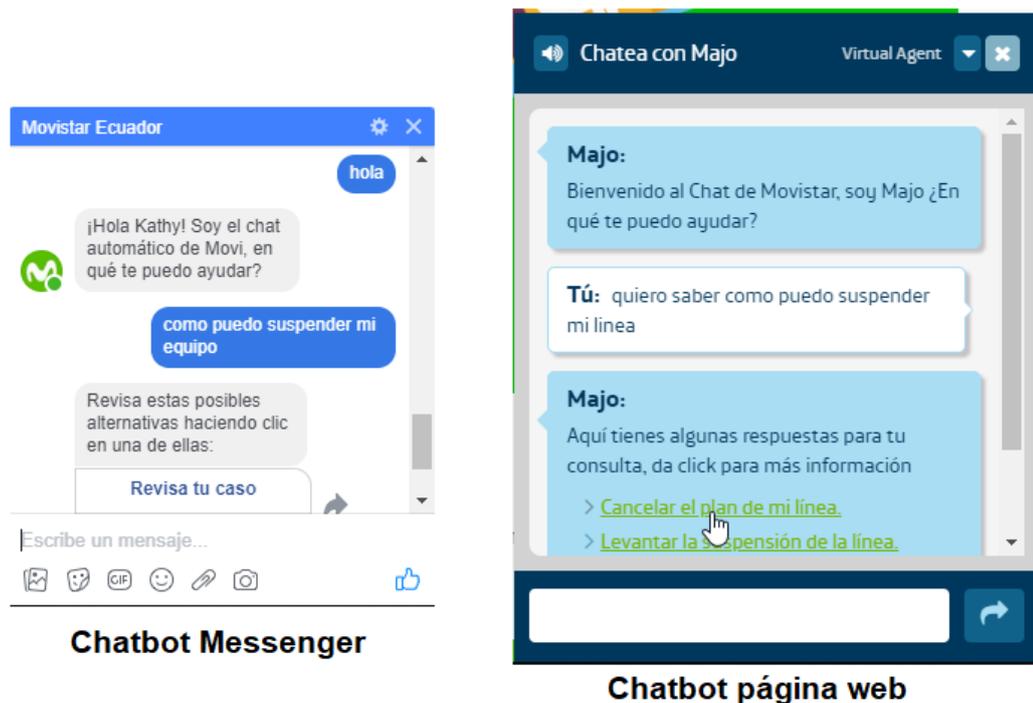


Ilustración 11. Servicio Movistar Ecuador domingo en la noche

Son muy pocas páginas ecuatorianas las que cuentan con servicio de asesor virtual, es difícil establecer estadísticas y mucho menos normativas de regulación. Por esta razón, los asistentes virtuales son una opción muy atractiva en el mercado ecuatoriano y además brindan una herramienta adicional en la mejora del servicio al usuario.

2.2 Herramienta para el Desarrollo de Chatbots

2.2.1 Tecnologías para el desarrollo de Chatbots

En la actualidad, las grandes empresas de software ven a las interfaces conversacionales como una evolución del uso del software a través de internet, ofreciendo soluciones que proporcionen una infraestructura que facilite el desarrollo de aplicaciones basada en IA. [23]

Para la creación de Chatbots existen varias herramientas, la gran mayoría son de pago y cuentan con licencia para distintos usos.

Entre las más conocidas están:

- **Microsoft Bot Framework:** Se trata de un framework de licenciado por Microsoft para la construcción y administración chatbots a través de distintos canales como son web y hardware especializado, cuenta además con la integración de los servicios cognitivos de Azure. [24]

- **IBM Watson Conversation:** framework famosos desarrollado por IBM para la creación y testeo de ChatBots en distintos canales, dispositivos móviles, tarjetas Arduino e incluso robots. Este framework cuenta con un kit de desarrollo apoyado en los servicios cognitivos de Watson y acompañado de IBM cloud. [25]
- **Bot Libre:** éste cuenta un kit de desarrollo propietario que posibilita el acceso a su API web desde JavaScript, Android, Java, iOS y Objective C, también cuenta con acceso a descarga del código fuente y librerías JAR desde GitHub y Sourceforge. Como ventaja también cuenta con chatbots animados. [26]
- **Wit.ai:** Se trata de un servicio web para desarrolladores en donde se encuentran herramientas para la creación de apps que permiten hablar o enviar mensajes de texto para Facebook. [27]
- **DialogFlow:** es un api para la creación ChatBots. Esta ofrece a los usuarios nuevas formas de construcción de interfaces conversacionales de voz y texto basadas en inteligencia artificial.

DialogFlow tiene la ventaja de poseer compatibilidad con Google Assistant, Amazon Alexa, Facebook Messenger y a muchas otras plataformas y dispositivos populares.

Cuenta además con la posibilidad de crear flujos con palabras clave, te permite establecer normas de reconocimiento natural del lenguaje y varias respuestas naturales. [5]

Tabla 2. Comparativa de herramientas

		Herramienta				
		Microsoft Bot Framework	IBM Watson Conversation	Bot Libre	Wit. Ai	DialogFlow
Características	Pagado	sólo versión premium	SI	NO	NO	sólo versión premium
	Requiere integraciones para PLN	SI	NO	NO	NO	NO
	Es multiplataforma	SI	SI	SI	SI	SI
	Soporte en varios idiomas	SI	SI	NO	SI	SI
	Ofrece reportes de uso de la herramienta	SI	SI	NO	NO	SI
	Permite Entrenamiento por el usuario	SI	SI	SI	SI	SI

Permite carga de conocimiento a través de archivos	SI	NO	NO	SI	SI
Reconocimiento de temas o tópicos de la conversación	Intents	Intents	Intents	Historias	Intents
Se puede modificar la configuración en tiempo real	NO	SI	NO	NO	SI
SDKs disponibles	Limitados	SI	NO para NODE JS	SI	SI
Dispone de consola de administración	SI	SI	NO	SI	SI
Reconocimiento de voz	SI/Pagado	SI/Pagado	NO	SI	SI/Pagado
Curva de aprendizaje	ALTA	MEDIA	ALTA	ALTA	MEDIA

Fuentes de la tabla 2: [28] [26] [29] [30] [31] [32] [27] [33] [34]

En el mercado existen muchas alternativas para la creación del Chatbot, cada herramienta tiene alguna característica que la diferencia de otra, se han evaluado las características en función de las necesidades del negocio donde la principal es utilizar una herramienta libre y fácil de entender ya que el personal que se hará cargo de dar soporte a la solución deberá realizar cambios y configuraciones en el entorno de la herramienta por lo que se decidió utilizar DialogFlow antes conocido como API.AI. El motivo de la utilización de DialogFlow fue el uso intuitivo de la herramienta ya que cuenta con una consola de administración gráfica y fácil de usar, además de contar con una licencia gratuita y una interfaz de reportes de uso de la herramienta, al que también se debe resaltar es la amplia documentación y disponibilidad de SDKs para la integración con varias plataformas como PHP y NodeJS.

Otra cosa que se tomó en cuenta es la curva de aprendizaje de todas las plataformas, donde la curva de aprendizaje de DialogFlow es menor a la de las demás herramientas; los autores de la solución se han visto involucrados experimentalmente para definir el resultado de esta característica.

2.3 Metodologías de Desarrollo de Software

El uso de metodologías tiene el objetivo de lograr la construcción de un sistema eficiente, que cumpla con los requerimientos necesarios para su correcto funcionamiento. Las metodologías de desarrollo sugieren el uso de un proceso disciplinado con el fin de hacerlo más predecible y eficiente.

El principal objetivo del uso de una metodología de desarrollo es aumentar la calidad del software durante sus fases de desarrollo. [35]

2.3.1 Metodologías de desarrollos tradicionales vs ágiles

No existe una metodología de desarrollo software que pueda ser ocupada en todos los casos, esto es debido a las distintas necesidades existente durante el desarrollo de un sistema, esto demanda que la metodología seleccionada sea adaptada a las características de cada proyecto (equipo de desarrollo, recursos, etc.)

Las metodologías de desarrollo se agrupan en dos grupos con similares características y objetivos estas son: ágiles y robustas. [35]

2.3.2 Metodologías Ágiles de desarrollo de software

Tomando en cuenta las necesidades del desarrollo del sistema se hará uso de las metodologías ágiles para cumplir con la mayoría de los requerimientos del stakeholder y por el tiempo necesario para el cumplimiento de este junto con la necesidad de cambios durante la presentación de los prototipos.

Además, la metodología Ágil se define oficialmente en su manifiesto con las siguientes características:

- Individuos e interacciones sobre los procesos y herramientas
- Software funcional sobre la documentación extensa
- La colaboración del stakeholder sobre la negociación contractual
- La respuesta sobre cambios antes del seguimiento del plan

Se han definido los siguientes parámetros para que el uso de la metodología cumpla con los requerimientos y ofrezca un producto de mejor calidad:

- **Rápida Implementación:** el stakeholder del proyecto desea mejorar la productividad de respuesta a las necesidades de sus usuarios lo más pronto, esto genera la necesidad de entregar prototipos funcionales.
- **Participación del cliente:** Durante este proyecto el stakeholder solicita su participación con la idea de mantener una comunicación constante con el equipo de trabajo y poder dar seguimiento, por lo cual es necesario que el stakeholder sea integrado en las actividades del proyecto
- **Requerimientos imprecisos:** durante el desarrollo comienzan a aparecer nuevas necesidades debido a que el stakeholder analiza

problemas recurrentes y define las necesidades de sus usuarios a través de etapas (ejemplo llenado de formularios con opciones múltiples).

- **Equipo de trabajo de dos personas:** durante el desarrollo es necesario administrar la carga de trabajo, horarios de reunión y tiempo empleado en el desarrollo y documentación, la metodología de desarrollo debe ser una herramienta en la asignación de responsabilidades y roles.
- **Proyectos que se han realizado aplicando la metodología:** se debe tomar en cuenta la experiencia del equipo de desarrollo sobre el uso de metodologías ágiles, ya que esto permite que obtener resultados con más expertos en términos de eficiencia y eficacia.
- **Información de soporte disponible:** Es necesario contar con documentación sobre la aplicación de metodologías para que estas puedan ser guías para los miembros del equipo.

2.3.3 Metodología XP y SCRUM

La metodología según Kent Beck precursor en el uso de metodologías ágiles, “XP fue desarrollada para el cumplimiento de las necesidades específicas durante el desarrollo de software que tiene la característica de ser realizado por equipos de 2 hasta 10 programadores y que durante el desarrollo necesitan realizar cambios constantes en los requisitos...”. Adicionalmente cuenta con parámetros de gestión de cambio, gestión de pruebas y enfoque de desarrollo usando prototipos.

Kent Beck asevera que el uso de esta metodología permitiría que las pruebas pueden ser ejecutadas en una fracción de un día. Por estas razones, el equipo de trabajo estima que XP es una alternativa viable para el desarrollo de este proyecto. [36]

Durante la definición de metodología el equipo de trabajo considero la necesidad contar con una herramienta de gestión del proyecto. Por esta razón, se ha seleccionado a Scrum ya que este framework ofrece una capacidad adaptativa y también propone la obtención de productos con calidad.

La toma de decisiones está basada en la experiencia del equipo y de la información disponible del proyecto. SCRUM tiene tres pilares en los que se basa para el desarrollo del proyecto [37]:

- **Transparencia:** todos los responsables deben conocer los aspectos, viabilidad y condiciones del proyecto.
- **Inspección:** los miembros deben realizar un seguimiento con el uso de los artefactos de Scrum para la detección de cualquier inconveniente para darle solución de manera oportuna y se pueda llegar a las metas propuestas.

- **Adaptación:** En caso de que exista una desviación durante el desarrollo que terminara en un producto no aceptable o en problemas de difícil solución, el equipo deberá tomar un proceso de mitigación del riesgo y reformar lo necesario para el cumplimiento de los requerimientos.

2.3.4 Artefactos de Scrum

Los artefactos de Scrum deben estar disponibles el equipo del proyecto, estos deben permitir la detección y recuperación en caso de existir desviaciones en los objetivos del proyecto [37].

- **Pila de productos (Product Backlog)**, es un listado que contiene los requerimientos necesarios que debe contener el producto final. En primera instancia solo contiene los requerimientos más generales y, conforme el proyecto avanza debe cumplir con la visión y expectativas del cliente o stakeholder.
- **Pila de sprint (Sprint Backlog)**, Es un conjunto de ítems del Product Backlog que se deben realizar durante una iteración. En caso de que la planificación del Sprint no sea adecuada o si se existe un indicio de que no se cumplirá el objetivo, el sprint backlog deberá ser modificado.
- **Gráfico de Trabajo Pendiente (Burndown Chart) o incremento**, este artefacto permite la evaluación durante el tiempo del cumplimiento de objetivos y de las tareas pendientes.

2.3.4.1 Scrum Team

El Scrum Team está constituido por [37]:

- **Dueño del Producto (product owner):** Es responsable de brindar las facilidades y requisitos del producto. Para este proyecto será la Ing. Tania Gualli en representación de la DGIP.
- **Scrum Master:** Es el responsable de que el Equipo Scrum se adhiera a la teoría de Scrum, prácticas y normas. Para el proyecto será el director del proyecto de titulación.
- **El Equipo de Desarrollo (SCRUM team):** Está conformado por los autores del proyecto encargados de hacer la entrega del producto al final de cada Sprint.

La combinación del framework SCRUM y la metodología XP satisface a cabalidad las características del proyecto; con esto se busca entregar al cliente productos de software de calidad, que mejoren la eficiencia y eficacia de los procesos que se realizan actualmente, además de brindar una herramienta tecnológica a la comunidad politécnica.

2.4 Tecnologías Utilizadas

El proyecto consta de un sistema web y una aplicación móvil con tecnología web. Durante el desarrollo se tomó la decisión de utilizar como base la tecnología web, debido a las limitaciones tecnológicas en los dispositivos ubicados en los kioscos dentro de la institución. Y para el sistema web se desea la implementación de un plugin para el WordPress de la página institucional, adicionalmente se solicitó una página para la revisión de estadísticas de las apps.

2.4.1 Tecnología y herramienta de desarrollo para la implementación del servidor

De acuerdo con la arquitectura definida y aceptada por el cliente se realiza la lógica del servidor web con Node JS como backend.

2.4.1.1 Node JS

Node JS es un ambiente de servidor de código abierto, es gratuito, es compatible con varias plataformas y lo que más resalta de esta herramienta es que se escribe el código del servidor en JavaScript, esto da más flexibilidad a las operaciones a realizarse del lado del servidor. [38]

Node.js en sí mismo es una implementación en C ++ de un motor V8 que permite al hardware interpretar las instrucciones escritas en JavaScript facilitando así la programación del lado del servidor y las aplicaciones de red, esta implementación es utilizada por el popular navegador web de Google, Chrome. [39]

A continuación, la ilustración 12 muestra la arquitectura de la de NodeJS, donde a diferencia de Chrome se tienen las librerías de Node JS, el Motor V8 para la interpretación a lenguaje de máquina y Livub C que trabaja con el procesamiento de entradas y salidas de forma asíncrona. [40]

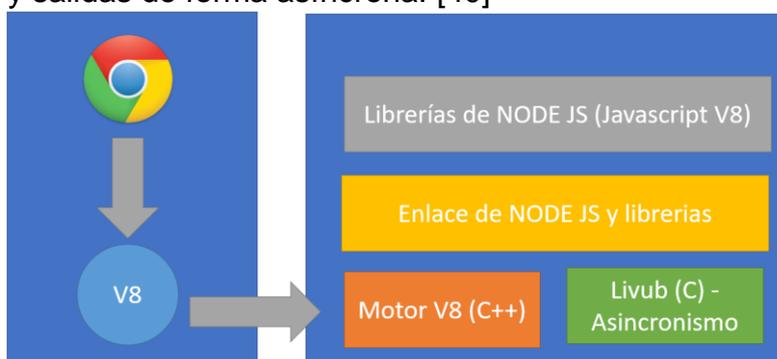


Ilustración 12. Arquitectura de NodeJS

Fuente [40]

Otra de las ventajas de NodeJS es que a diferencia de otros lenguajes como por ejemplo PHP, Node JS trabaja de forma asíncrona, sin bloqueos de entrada/salida y se ejecuta en un solo hilo de procesamiento esto lo hace más liviano y eficiente. Por ejemplo, para leer un archivo en PHP se tienen que seguir los siguientes pasos:

1. Envía la tarea al sistema de archivos de la computadora.
2. Espera mientras el sistema de archivos se abre y lee el archivo.
3. Devuelve el contenido al cliente.
4. Listo para manejar la siguiente solicitud.

En Node JS esto se reduce a los siguientes pasos:

1. Envía la tarea al sistema de archivos de la computadora.
2. Está listo para manejar la siguiente solicitud.
3. Cuando el sistema de archivos se abre y lee el archivo, el servidor devuelve el contenido al cliente.

Node JS elimina la espera y continua con la siguiente solicitud lo que le permite manejar más operaciones. [38]

Node JS es un entorno de ejecución muy flexible y de fácil escalada ya que cuenta con su propio gestor de paquetes el conocido NPM (Node Package Manager por sus siglas en inglés), este gestor tiene una gigantesca cantidad de paquetes de gran utilidad para el desarrollo del servidor con Node JS, va desde un paquete para impresión de logs hasta un framework para la gestión de peticiones https. Además, varias de las APIs que permiten el uso de asistentes virtuales tienen un paquete de integración con NodeJS como por ejemplo DialogFlow.

Como se ha descrito en los párrafos anteriores, Node JS tiene muchas ventajas al momento de utilizarlo del lado del servidor, razón por la cual se ha utilizado este entorno para la implementación de la lógica del servidor de los asistentes virtuales del presente proyecto.

Java 8

Java es un lenguaje de programación y una plataforma informática creada por Sun Microsystems. En la actualidad muchas aplicaciones y sitios web dependen de JAVA para su funcionamiento. Java se considera un lenguaje de programación rápido, seguro y fiable. Su utilización va desde el uso doméstico como son teléfonos móviles y videojuegos hasta grandes servidores de datos que son usados por grandes empresas. [41]

Actualmente, JAVA se encuentra en su versión 8 la que incluye nuevas características, mejoras y correcciones de bugs para mejorar la eficacia en el desarrollo y la ejecución de programas Java. Esta versión ya pasa su fase de pruebas y certificaciones para otorgar una versión estable que pueda ser usada por la mayoría de los usuarios. [41]

A continuación, un resumen de las mejoras incluidas en Java 8: [41]

- **API de fecha y hora**
Esta nueva API permite a los administradores gestionar de manera sencilla los datos de fecha y hora.
- **Motor de JavaScript Nashorn**
Esta nueva implantación permite aligerar el rendimiento del motor de

JavaScript integrado dentro del JDK y está disponible para su uso mediante APIs.

- **Métodos de extensión virtual y expresión Lambda**
Una de las funciones destacables de Java SE 8 es la implantación de expresiones Lambda y funciones adyacentes a la plataforma y el lenguaje de programación Java.
- **Seguridad mejorada**
Sustitución de métodos usados por el emisor dentro de una comunicación por métodos mejorados con nuevos mecanismos más fiables y fáciles de implementar

Dentro del proyecto se hizo uso de JAVA para crear un motor que nos permita comunicar la app web con JasperSoft el cual nos genera un archivo Jasper con el cual se crea el documento con los datos obtenidos en el formulario web implementado dentro de uno de los contextos del chatbot. El motor JAVA es el encargado de gestionar la comunicación entre los aplicativos y de organizar, modificar,

2.4.2 Tecnología y herramienta de desarrollo para la construcción del SmartBot

2.4.2.1 Desarrollo del plugin para WordPress

Según un estudio realizado por W3Techs, más de un tercio de las páginas web en el mundo hacen uso de WordPress. Básicamente su éxito depende de la facilidad que brinda este software para la creación y gestión de los sitios web. []

WordPress es gratuita y su fuente está abierta al público, debido a que su creador Matt Mullenweg lo declaró como un proyecto de software libre con la intención de garantizar su continuidad, gracias a la comunidad de miles de desarrolladores voluntarios el proyecto ha logrado posicionarse como el sistema de gestión de contenido (CMS “Content Management System”) predilecto por desarrolladores y no desarrolladores, ya que WordPress cuenta con un sistema intuitivo que permite configurar una página web de manera visual. [42]

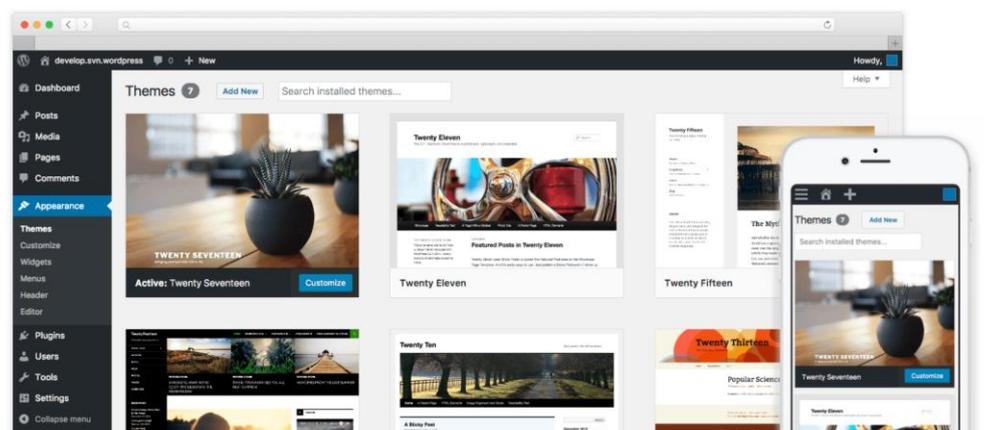


Ilustración 13. Temas WordPress Fuente: <https://es.wordpress.org/>

WordPress cuenta con bastante adaptabilidad de lenguajes de programación. Su principal Lenguaje es PHP esto se debe a que cerca del 85% de la web obtiene sus datos del lado del servidor. [42]

Como gestor de contenidos web también cuenta con HTML, CSS y JavaScript. HTML es utilizado para crear el esqueleto del sitio web, en la actualidad se cuenta con HTML5 el cual brinda nuevas características sobre todo en el aspecto de multimedia. CSS es el encargado de brindar al aplicativo los estilos necesarios para que la web page se vea más atractiva visualmente. JavaScript, por otra parte, se encarga de brindar características interactivas como son los carruseles con imágenes que le dan al aplicativo dinamismo. [42]

WordPress también hace uso de MySQL como base de datos. MySQL es muy útil para el gestor de contenido, debido a que cuenta con un registro del uso del web page, lo cual facilita saber cuáles fueron las entradas más importantes por categoría, esto ayuda con el balance de carga que se refleja en el rendimiento del aplicativo. [42]

Adicionalmente a esta variedad de posibilidades WordPress cuenta con una infinidad de plantillas que facilitan el trabajo de programación y desarrollo visual durante la creación de la web page.

Para el desarrollo del proyecto fue necesario hacer uso de las características de HTML5, CSS y JS para la creación de la interfaz de comunicación entre el usuario y el Bot, además se debió hacer uso de características multimedia que permitan la animación y la interacción con el usuario.

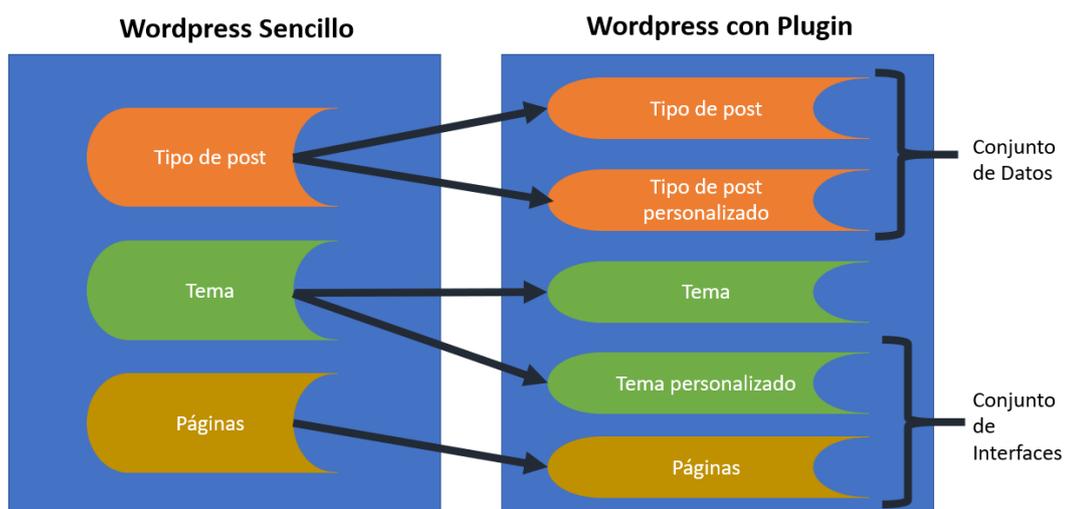


Ilustración 14. implementación del plugin en WordPress

Adicionalmente fue necesario crear un enlace con los reportes de uso de los aplicativos, la aplicación de reportes hace uso de Angular para poder generar los datos necesarios para ser presentados al usuario en este caso los administradores de la web page WordPress y el aplicativo cargado en los diferentes kioscos ubicados dentro de la institución.

2.4.2.2 Desarrollo App para su ejecución en los kioscos de la EPN

El desarrollo de aplicaciones móviles debe enfrentar el actual problema de la fragmentación de mercado e inclusive dentro de la misma marca la variación de tecnología. En Ecuador no se ha realizado estudios sobre el uso de dispositivos móviles y sus sistemas operativos por parte de entidades como MINTEL e INEC; sin embargo, estas instituciones poseen estadísticas sobre el aumento en uso de móviles en el país.

Para el caso de estudio se estimó el desarrollo para dos sistemas operativos móviles como son Windows Phone y Android cada uno con su asistente, pero a nivel mundial se encuentra Android como principal SO como se muestra en la gráfica, por tanto se toma en cuenta el desarrollo de mercado que Android ha tenido a nivel mundial, para el actual proyecto se limita el desarrollo a Android por factibilidad de dispositivos dentro de la entidad, pero no se cierra la posibilidad de desarrollo a futuro sobre otros SO.

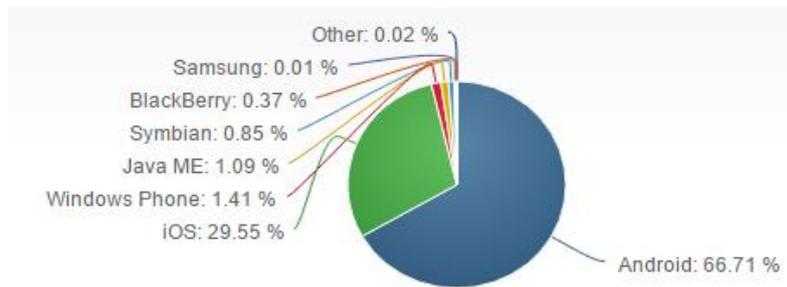


Ilustración 15. Cuota del mercado de sistemas operativos móviles

Fuente: [43]

Android

Es un sistema operativo móvil basado en el núcleo de Linux presentado en 2007, cuenta con muchos dispositivos inteligentes, y se han creado millones de aplicaciones para uso personal o corporativo [44]. En la ilustración se puede observar la Arquitectura de Android:

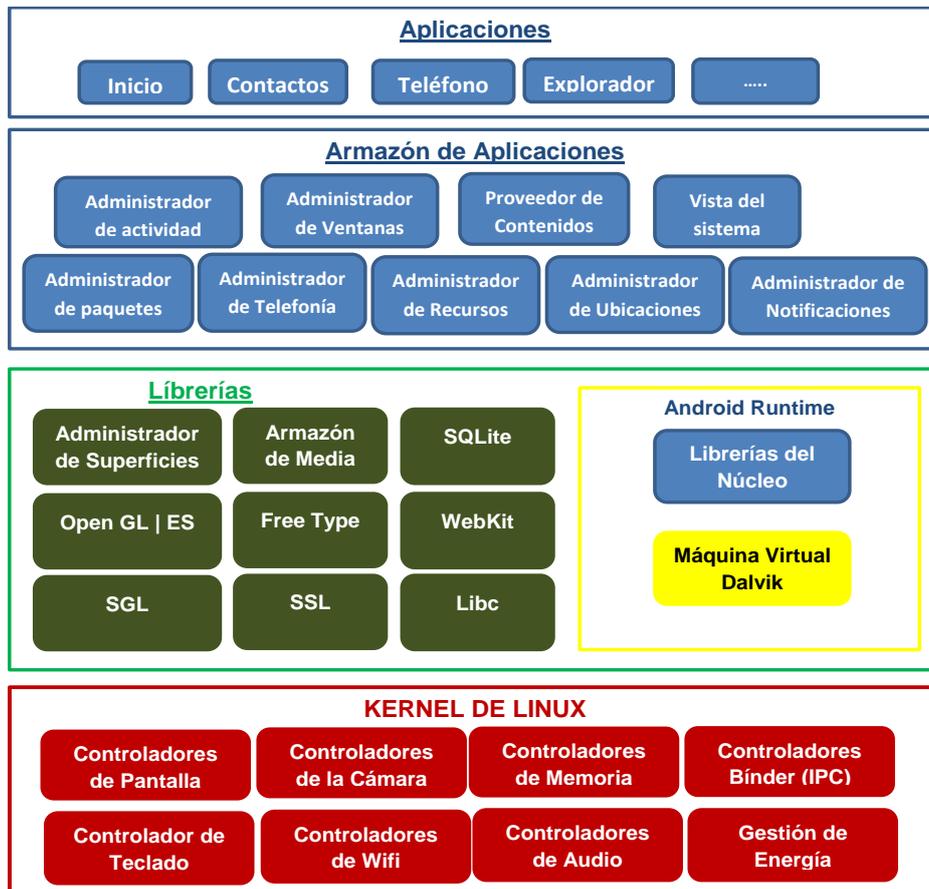


Ilustración 16. Arquitectura de Android

Fuente. [44]

SENCHA

Sencha Touch es un marco de trabajo para el desarrollo de aplicativos móviles centrado en WebKit (base de los navegadores web Safari, Google Chrome). Sencha es un entorno basado en HTML5, JS y CSS3. La apariencia de los aplicativos generados son similares a las que se obtienen nativamente en Android, BlackBerry y iOS.

Sencha cuenta con licencia del tipo comercial y Open Source GPLv3.

Como principal ventaja tiene la cantidad de controles IU o elementos de interfaz Sencha fue diseñado específicamente para dispositivos táctiles por lo que cuenta con una gran cantidad de eventos táctiles o gestos.

Además, con nuevos eventos como son: tap, double tap, tap and hold, swipe, rotate o drag and drop. [45]

Capítulo 3 Desarrollo

3.1 Definición del proyecto

El objetivo principal de este proyecto de tesis fue desarrollar un asistente virtual como recurso de apoyo que facilite el manejo de las demandas que tiene la DGIP por parte de los usuarios de la EPN, para la creación del asistente integrado en la página WordPress de la EPN y el asistente dirigido por voz para los kioscos fue necesario utilizar tecnologías de procesamiento de lenguaje natural (PLN) y reconocimiento de voz (STT).

El desarrollo cuenta con la implementación de un agente de procesamiento de lenguaje natural y machine learning para resolver las inquietudes en forma de texto, este agente hace uso del API de DialogFlow para generar las respuestas, adicionalmente fue necesario implementar un servidor (webhook) encargado de direccionar las peticiones de los usuarios a las diferentes APIs y registrar el uso de los asistentes.

3.1.1 Características del proyecto

Se realizaron varias entrevistas con el grupo de trabajo antes de comenzar con el desarrollo, durante estas entrevistas se trataron ciertas características que debe tener este proyecto, estas son:

- El grupo de desarrollo deberá interactuar con el interesado en el proyecto para evaluar las necesidades del sistema en periodos cortos.
- El grupo de desarrollo deberá estimar los tiempos y esfuerzos necesarios para la visualización de resultados y finalización del proyecto.
- El grupo de desarrollo estimara los posibles riesgos de cambios imprevistos dentro de la etapa de desarrollo del proyecto.
- El grupo de desarrollo deberá obtener cualidades de trabajo en equipo y deberá dividir el trabajo para este proyecto entre los dos involucrados
- El grupo de desarrollo y los interesados deberán definir los distintos reglamentos de confidencialidad.
- El grupo de desarrollo y los interesados deberán definir los sitios de trabajo y la manera de obtención de credenciales necesarias para el desarrollo del proyecto.

3.1.2 Requerimientos definidos por el equipo de trabajo

- Desarrollo de un Plugin compatible con WordPress para ser usado dentro de la página Web institucional, el plugin debe ser capaz de brindar respuestas en texto y usar recursos multimedia.
- Un ChatBot que resuelve preguntas con solicitud web y sea capaz de interconectar el servicio de DialogFlow con el aplicativo web.
- Un ChatBot que resuelve las preguntas generadas en los kioscos de la EPN y sea capaz de interconectar el servicio de DialogFlow con el aplicativo.
- Desarrollo de un aplicativo web que contenga una solución para el llenado de formularios.

3.1.3 Selección de la metodología de desarrollo

En la tabla 3 se definen características necesarias para el proyecto que poseen las metodologías tradicionales y ágiles explicadas en el capítulo anterior. Las metodologías tradicionales cuentan con la definición de los requerimientos durante el inicio del proyecto, pero esta característica es parcialmente necesaria dada la necesidad de que el stakeholder o interesado participe durante el desarrollo.

Tabla 3. Comparativa metodología tradicional vs ágil

Fuente:[35]

Características	Necesidades del proyecto		
	no necesario	parcial	prioridad
	1	2	3
Tradicionales			
Definición de los requerimientos del sistema al inicio del proyecto		X	
Verificación final de la planificación establecida		X	
Colaboración con el stakeholder solo en etapas iniciales y finales		X	
Comunicación formal con el stakeholder	X		
Testeo del sistema en la implementación			X
Modificaciones en el proyecto son muy costosas en tiempo y esfuerzo	X		
La necesidad de dividir en muchos roles durante el proyecto	X		
Total	12		
Ágiles			
Trabajo conjunto con el cliente durante el proyecto			X
Feedback y cambios de requerimientos durante el proyecto			X
Comunicación con el cliente de manera informal		X	
Probar el sistema durante el desarrollo del proyecto (entregables funcionales)			X
Cambios durante el proyecto son más sustentables		X	
Definición de solo roles necesarios para el desarrollo		X	
Total	15		

Características como el testeo o verificaciones de las etapas de desarrollo del proyecto son necesarias para su implementación en un entorno de producción, sin embargo, para la metodología tradicional se debe tomar en cuenta, que los cambios de un sistema en etapas finales son mucho más costosos en tiempo y esfuerzo de realizar.

Para las metodologías ágiles se toma en cuenta mayormente sus características de trabajo en equipo junto con el interesado lo que facilita la comunicación y permite que las correcciones se realicen de manera oportuna. Los entregables funcionales facilitan la deducción de la factibilidad de un requisito y las posibles mejoras que esta pueda tener.

La elaboración de la tabla 3 permitió determinar ciertas características necesarias para el proyecto de desarrollo del asistente virtual y se realizó una evaluación entre las características brindadas por las metodologías tradicionales y ágiles expresada en la tabla 3 llegando a la conclusión de que las metodologías ágiles son las indicadas para el desarrollo proyecto.

Se hizo uso de metodologías como son: Scrum y Xtreme Programming durante el desarrollo del proyecto.

3.2 Evolución del proyecto

Una vez establecido las herramientas, tecnologías y las metodologías de desarrollo; se continuó con la recopilación de las historias de usuario establecidas en la metodología de desarrollo XP, se estimó el tiempo y los costos del desarrollo del proyecto. Las historias de usuario fueron basadas en las funcionalidades que los usuarios esperaban tener y estas ayudaron a facilitar el testeo del sistema [47].

Se hizo uso del product backlog para listar las funcionalidades a implementar, cada funcionalidad fue descrita por su contenido, disponibilidad y prioridad. Este artefacto SCRUM facilito realizar el balance entre las tareas completadas y las tareas a realizarse.

3.2.1 Historias de Usuario Épicas

Para poder cumplir con los requerimientos fue necesario generar las Historias de usuario épicas. Estas se muestran a continuación:

Tabla 4 Historia épica HUE1

HUE1 - Consultar información con respecto a contenido Académico	
Usuarios: Estudiantes – Docentes	
Prioridad de negocio: Alta	Riesgo de desarrollo: Alto
Descripción: Como usuario quiero poder preguntar al asistente sobre temas correspondientes a contenido académico ej. Carreras, guías de configuración, etc.	
Observaciones: Estos contenidos se presentarán en ambos canales como son móvil y web, se debe tomar en cuenta la necesidad de usar elementos multimedia.	

Tabla 5 Historia épica HUE2

HUE2 – Obtener respuestas con tecnologías TTS y STT	
Usuarios: Estudiantes – Docentes	
Prioridad de negocio: Alta	Riesgo de desarrollo: Alto
Descripción: Como usuario quiero obtener un resultado hablado a las preguntas hechas al asistente mediante el uso de voz como fuente a la pregunta realizada	
Observaciones: Este contenido multimedia será único para el canal móvil y se presentará con el uso prioritario de tecnología libre.	

Tabla 6 Historia épica HUE3

HUE3 - Consultar información con respecto a estadísticas de uso de los asistentes	
Usuarios: Administrador	
Prioridad de negocio: Medio	Riesgo de desarrollo: Medio
Descripción: Como administrador quiero poder visualizar el nivel de uso de los asistentes virtuales y las preguntas que no obtuvieron una respuesta adecuada.	

Observaciones:

Se el contenido solicitado se mostrará en una plataforma web.

Tabla 7 Historia épica HUE4

HUE4 – Realizar el llenado del formulario RM-525	
Usuarios: Estudiantes	
Prioridad de negocio: Media	Riesgo de desarrollo: Medio
Descripción: Como usuario quiero poder realizar la generación de un documento necesario para la reinscripción de materias dentro de la facultad, ingresando la información requerida para que se efectué este proceso.	
Observaciones: Se hará uso de una ventana para el llenado de este documento y este será generado exclusivamente para el asistente web.	

3.2.1.1 Product Backlog y Release Planning

En la tabla 8, se describe las historias de usuario específicas para cada historia de usuario épica, a cada historia de usuario se le asigna una prioridad y un detalle sobre la funcionalidad de cada HU. Para la generación del release planning fue necesario asignar las actividades a cada sprint.

Tabla 8 Product Backlog

HUE	HU	Usuarios	Funcionalidad	Móvil	Web	Prioridad	Sprint	Duración (Días)
HUE01	HU01	Estudiante Docente	Desplegar botón de apertura		X	5	0	20
	HU02	Estudiante Docente	Visualizar lista de opciones	X	X	5	0	3
	HU03	Estudiante Docente	Manejar frases inadecuadas	X	X	3	1	1
	HU04	Estudiante Docente	Desplegar respuestas para cada opción	X	X	5	1	2
	HU05	Estudiante Docente	Presentar información en Carruseles	X	X	2	1	4
	HU06	Estudiante Docente	Desplegar documentos en ventanas	X	X	3	1	2
	HU07	Estudiante Docente	Presentar opción de llenado de formulario		X	5	1	2

	HU08	Estudiante Docente	Abrir recorrido virtual de la EPN	X	1	1	1
HUE02	HU09	Estudiante Docente	Captar voz del usuario	X	5	2	5
	HU10	Estudiante Docente	Convertir voz en texto	X	5	2	1
	HU11	Estudiante Docente	Presentar Respuesta en la interfaz	X	5	3	3
	HU12	Estudiante Docente	Descartar Simbologías no necesarias	X	2	2	1
	HU13	Estudiante Docente	Emitir respuesta mediante sintetizador de voz	X	5	3	10
	HUE03	HU14	Administrador	Obtener interfaz de selección de opciones	X	5	4
HU15		Administrador	Presentar estadísticas de asistente móvil	X	4	4	10
HU16		Administrador	Presentar estadísticas de asistente web	X	4	4	2
HUE04	HU17	Estudiante	Presentar formulario	X	5	5	4
	HU18	Estudiante	Generar documento formulario lleno	X	5	5	20
				TOTAL, DIAS	96		

Para el Sprint 0 se estimó realizar la configuración del servidor y el levantamiento de los servicios necesarios para la funcionalidad de los asistentes.

En la tabla 9 se resume la distribución de las historias de usuario para cada uno de los Sprints

Tabla 9 Resumen Release Planning

Sprint 0	Sprint 1	Sprint 2	Sprint 3	Sprint 4	Sprint 5
HU01	HU03	HU09	HU12	HU14	HU17
HU02	HU04	HU10	HU13	HU15	HU18
	HU05	HU11		HU16	
	HU06				
	HU07				
	HU08				

3.2.2 Controles

En la tabla 10, se muestra una serie de controles que permitió que el desarrollo del proyecto se realice más sistemáticamente para obtener los mejores resultados, estos han sido obtenidos de mejores prácticas de Scrum propuestas por Verman y Lacey. [48] [49]

Tabla 10 Tabla comparativa de controles

Controles	G. Burndown	T. de Criterio	Ctrl. Retrospectiva Semáforo
¿Por qué?	<ul style="list-style-type: none"> - Poder comparar la velocidad de trabajo - Saber si es necesario aumentar gente al equipo 	<ul style="list-style-type: none"> - El cliente necesita puntos de referencia para ver si el prototipo cumple con los requerimientos estipulados. 	<ul style="list-style-type: none"> - La necesidad establecer parámetros de comunicación para la mejora en el trabajo
¿Para qué sirve?	<ul style="list-style-type: none"> - Visualizar el avance del proyecto 	<ul style="list-style-type: none"> - Generar criterios de aprobación de los prototipos entregados, observaciones sobre los mismos 	<ul style="list-style-type: none"> - Definir puntos a mejorar antes de empezar otro sprint
¿Cómo?	<ul style="list-style-type: none"> - Registrando el trabajo realizado 	<ul style="list-style-type: none"> - Una tabla contiene las historias de usuario, junto con un criterio que debe ser calificado como aceptado o necesita de algún cambio 	<ul style="list-style-type: none"> - Mediante indicadores - Conclusiones obtenidas
¿Cuándo?	<ul style="list-style-type: none"> - Diariamente 	<ul style="list-style-type: none"> - Después de entregar el prototipo 	<ul style="list-style-type: none"> - Al finalizar la reunión con el cliente
¿Quién interviene?	<ul style="list-style-type: none"> - Grupo de desarrollo 	<ul style="list-style-type: none"> - Grupo de desarrollo - Cliente 	<ul style="list-style-type: none"> - Grupo de desarrollo - Cliente
Resultado	<ul style="list-style-type: none"> - Grafica estadística de avance del proyecto 	<ul style="list-style-type: none"> - Tabla 	<ul style="list-style-type: none"> - Tabla con indicadores - Uso de colores

3.2.3 Ejecución de los Sprints

3.2.3.1 Sprint 0

Para empezar el desarrollo del Sprint fue necesario definir las Historias de usuario, junto con una descripción y las tareas o actividades a realizarse como se presenta en el anexo G en las tablas HU01 y HU02, la Tabla 11 detalla los criterios de aceptación y las observaciones del cliente.

Tabla 11 Criterios de aceptación Sprint 0

Cod	Historia de Usuario	Criterio de aceptación	SI/NO	Observaciones
HU01	Desplegar botón de apertura	Framework instalado.	SI	
		Instalación de base de datos	SI	
		Configuración API	SI	
		Configuración de protocolos de comunicación entre el ChatBot	SI	
		Esquema y Scripts de las tablas a utilizarse en las bases de datos.	SI	
		Librerías de conexión SQL y comunicación API probados	SI	
HU02	Visualizar lista de opciones	Configuración de saludo	SI	
		Configuración de menú	SI	
		Diseño de interfaz web	SI	Se solicitó un cambio en cuanto a la forma de presentar los datos
		Diseño de interfaz móvil	SI	Presentación de mockups

Después de la ejecución del Sprint 0, el tablero Kanban se mostraba de la siguiente manera.

Tabla 12 Kanban del Sprint 0

Pendiente	En Desarrollo	En Testeo	Terminada
HU09	HU03		HU01
HU10	HU04		HU02
HU11	HU05		
HU12	HU06		
HU13	HU07		
HU14	HU08		
HU15			
HU16			
HU17			
HU18			

3.2.3.1.1 Prototipo Sprint 0

El prototipo diseñado para este sprint contuvo:

- La interfaz de presentación del Chatbot
- Lista con opciones



Ilustración 17. Prototipo de visualización

Botón diseñado para su uso en el plugin



Ilustración 18. Prototipo visualización Chatbot desplegando saludo

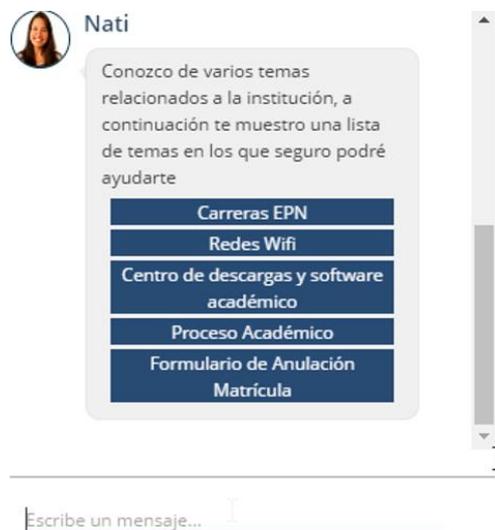


Ilustración 19. Prototipo visualización Chatbot listando menú de ayuda

3.2.3.1.2 Retrospectiva Sprint 0

Se obtuvo algunas demoras durante el desarrollo del sprint por desconocimiento del manejo del software, pero fueron remediadas en la etapa de desarrollo de las interfaces, lo cual permitió finalizar el sprint completando todas las actividades en el tiempo estipulado.

En la ilustración 20 se pudo visualizar la existencia de tareas que tomaron más tiempo del estimado pero el sprint 0 finalizó con 1 día antes de lo estimado lo que permitió más flexibilidad en los próximos Sprints, en tabla 13 se puede visualizar que uno de los parámetros con más inconvenientes fue la comunicación dentro del equipo.

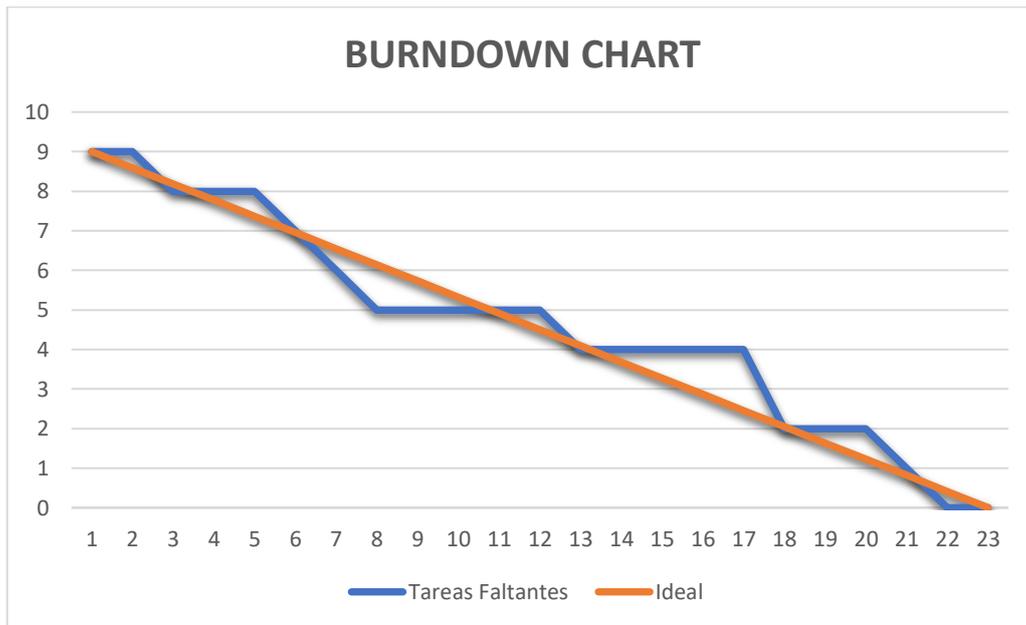


Ilustración 20. Burndown Chart Sprint 0

Tabla 13 Retrospectiva usando controles de mejoramiento Sprint 0

Parámetro	Calificación	Semáforo	Observaciones
Gestión de cambios	10	Verde	El uso de mockups permitió al cliente seleccionar de manera eficiente los detalles requeridos y el diseño
Comunicación con el cliente	10	Verde	Las reuniones se realizaron los días acordados y la entrega del prototipo se realizó con normalidad
Tareas cumplidas	10	Verde	Con algo de retraso se logró finalizar las actividades
Trabajo en equipo	5	Amarillo	falta de comunicación y de herramientas colaborativas

3.2.3.2 Sprint 1

El objetivo de este sprint fue crear la interacción visual con el usuario e implementar las frases de respuesta del ChatBot, las siguientes tareas o actividades a realizarse se presenta en el Anexo G en las tablas HU03 hasta la tabla HU08, en la Tabla 14 se detallaron y resumieron los criterios de aceptación y las observaciones dadas por el cliente.

Tabla 14 Criterios de aceptación Sprint 1

Cod	Historia de Usuario	Criterio de aceptación	SI/NO	Observaciones
HU03	Manejar frases inadecuadas	El ChatBot hace uso de la base de conocimiento	SI	
		El chat envía un dialogo sobre la frase inadecuada	SI	
HU04	Desplegar respuestas para cada opción	Distinción del tipo de solicitud	SI	
		Respuestas sobre carreras EPN	SI	
		Respuestas sobre Redes wifi	SI	
		Respuestas sobre Centro de descarga	SI	
		Respuestas sobre proceso académico	SI	
		Respuestas sobre el formulario fa-201	SI	
HU05	Presentar información en Carruseles	Carrusel para carreras	SI	
		Carrusel para software	SI	
HU06	Desplegar documentos en ventanas	Despliegue de documento	SI	
HU07	Presentar opción de llenado de formulario	Implementación ayuda llenado	SI	
HU08	Abrir recorrido virtual de la EPN	Información sobre EPN	SI	

Después de la ejecución del Sprint 1, el tablero Kanban se mostraba de la siguiente manera.

Tabla 15 Kanban del Sprint 1

Pendiente	En Desarrollo	En Testeo	Terminada
HU12	HU09	HU03	HU01
HU13	HU10	HU04	HU02
HU14	HU11	HU05	
HU15		HU06	
HU16		HU07	
HU17		HU08	
HU18			

3.2.3.2.1 Prototipo Sprint 1

El prototipo diseñado para este sprint contuvo:

- Las opciones del menú de ayuda del asistente
- Diálogos para manejo de frases inadecuadas
- Diálogos en caso de que no se posea un contexto a la pregunta realizada
- Presentación de los carruseles con los datos de carreras y software que posee la EPN
- Presentación de los documentos usando una ventana auxiliar
- Opción sobre recorrido de la EPN



Ilustración 21. Prototipo visualización de botones de selección de opciones



Ilustración 22. Prototipo presentación de datos mediante carruseles

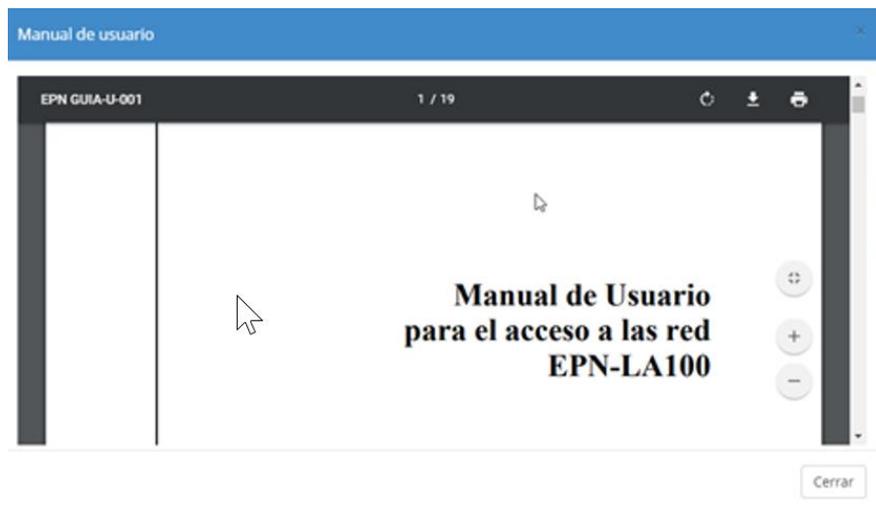


Ilustración 23. Prototipo visualización documento

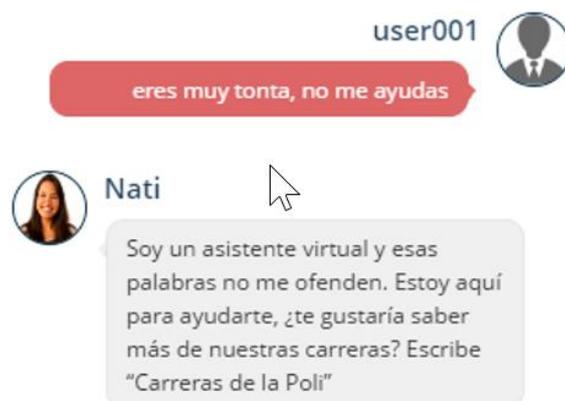


Ilustración 24. Prototipo de respuesta a frases inadecuadas para el asistente

Ilustración 25. Prototipo de formulario para generación de documento

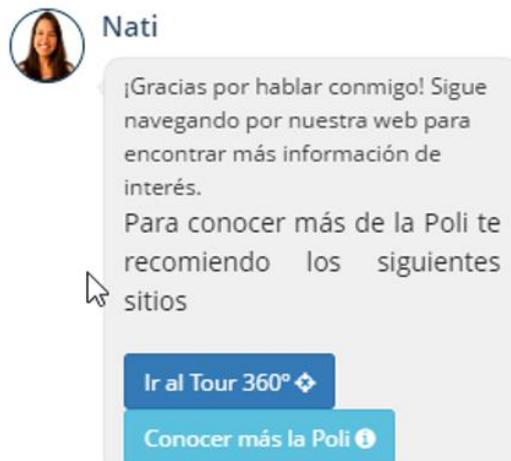


Ilustración 26. Prototipo de despedida del ChatBot

3.2.3.2.2 Retrospectiva Sprint 1

Para este sprint fue necesario reformular los contextos necesarios para la realización del proyecto y la reformulación de los elementos multimedia necesarios para mejorar la experiencia del usuario.

En la ilustración 27 se pudo visualizar la existencia de tareas que tomaron más tiempo del estimado a pesar de esto el sprint 1 finalizó de acuerdo con lo estimado, en tabla 16 se pudo visualizar que uno de los parámetros con más inconvenientes fue la gestión de cambio debido a cambios solicitados por el cliente.

BURNDOWN CHART

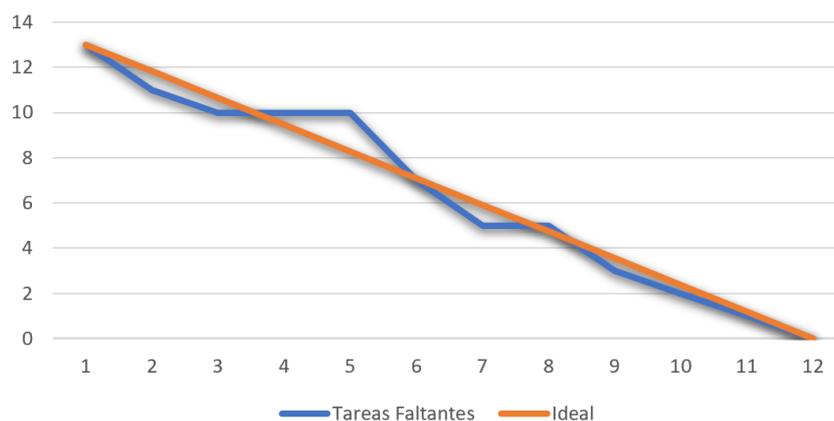


Ilustración 27. Burndown Chart Sprint 1

Tabla 16 Retrospectiva usando controles de mejoramiento Sprint 1

Parámetro	Calificación	Semáforo	Observaciones
Gestión de cambios	5	🟡	Para la realización de los cambios solicitados por el cliente fue necesario definir nuevamente la limitación de los contextos
Comunicación con el cliente	10	🟢	Las reuniones se realizaron los días acordados y la entrega del prototipo se realizó con normalidad
Tareas cumplidas	10	🟢	Todas las tareas se cumplieron a cabalidad
Trabajo en equipo	10	🟢	falta de comunicación y de herramientas colaborativas

3.2.3.3 Sprint 2

Para el desarrollo del Sprint fue necesario establecer las Historias de usuario las cuales son similares a las anteriores, fue necesario establecer las actividades a realizarse como se presentan en Anexo G en las tablas HU09 hasta la tabla HU11, en la tabla 17 se detallaron y resumieron los criterios de aceptación y las observaciones del cliente.

Tabla 17 Criterios de aceptación Sprint 2

Cod	Historia de Usuario	Criterio de aceptación	SI/NO	Observaciones
HU09	Captar voz del usuario	Captura de sonido	SI	Prueba en ambiente con ruido
		Icono representativo	SI	Minimalista
HU10		Texto dentro de un elemento visual	SI	

Cod	Historia de Usuario	Criterio de aceptación	SI/NO	Observaciones
	Convertir voz en texto	Integración a la interfaz	SI	
HU11	Presentar Respuesta en la interfaz	Texto de respuesta en un elemento visual	SI	
		Elementos multimedia	SI	

Después de la ejecución del Sprint 2, el tablero Kanban se mostraba de la siguiente manera.

Tabla 18 Kanban del Sprint 2 - Prototipo Sprint 2

Pendiente	En Desarrollo	En Testeo	Terminada
HU14	HU12		HU01
HU15	HU13		HU02
HU16			HU03
HU17			HU04
HU18			HU05
			HU06
			HU07
			HU08
			HU09
			HU10
			HU11

El prototipo diseñado para este sprint contuvo:

- Interfaz gráfica para la presentación del diálogo con el ChatBot
- Icono para representativo de voz
- Estilo para la aplicación Móvil



Ilustración 28 Prototipo visualización ChatBot móvil

3.2.3.3.1 Retrospectiva Sprint 2

Para este sprint fue necesario modificar la propuesta de ChatBot y cambiar de tecnología, pero se cumplió con los tiempos establecidos

En la ilustración 29 se pudo visualizar la existencia de tareas que tomaron más tiempo del estimado a pesar de esto el sprint 2 finalizó un día antes de lo estimado lo que permitió más flexibilidad en los próximos Sprints, en tabla 19 se pudo visualizar que uno de los parámetros con más inconvenientes fueron la gestión de cambio debido a cambios solicitados por el cliente y la comunicación con el mismo, esto debido a cambios en el personal del negocio del cliente.

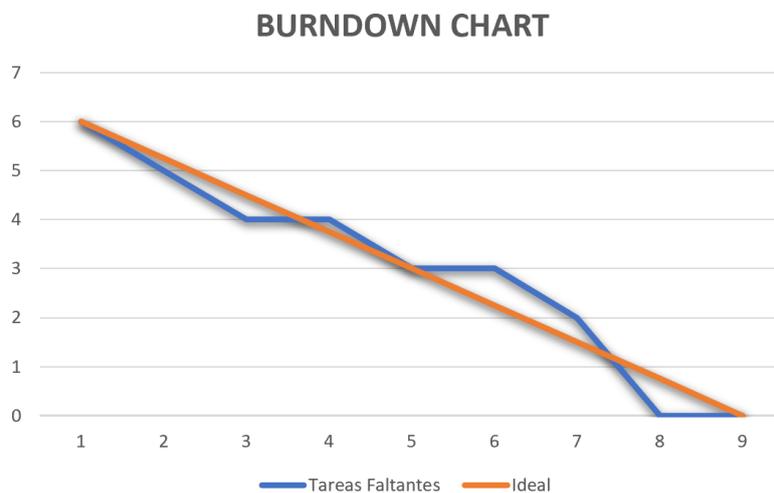


Ilustración 29 Burndown Chart Sprint 2

Tabla 19 Retrospectiva usando controles de mejoramiento Sprint 2

Parámetro	Calificación	Semáforo	Observaciones
Gestión de cambios	5		Fue necesario cambiar de tecnología de desarrollo por necesidades de adaptación con otras tecnologías implementadas por parte de la DGIP
Comunicación con el cliente	5		Las reuniones no se realizaron en las fechas establecidas, y los requisitos no fueron establecidos claramente
Tareas cumplidas	10		Con algo de retraso se logró finalizar las actividades
Trabajo en equipo	10		

3.2.3.4 Sprint 3

Este sprint se encargó de las tareas faltantes para finalizar la construcción del aplicativo móvil usando voz, las historias de usuario de este sprint fueron detalladas en el Anexo G en las tablas HU12 y HU13, en la Tabla 20 se detallaron y resumieron los criterios de aceptación y las observaciones del cliente.

Tabla 20 Criterios de aceptación Sprint 3

Cod	Historia de Usuario	Criterio de aceptación	SI/NO	Observaciones
HU12	Desplegar botón de apertura	Sin elementos multimedia	SI	
		Sin uso de signos especiales	SI	
HU13	Visualizar lista de opciones	Selección de características de voz	SI	
		Diferentes sintetizadores	SI	

Después de la ejecución del Sprint 3, el tablero Kanban se mostraba de la siguiente manera.

Tabla 21 Kanban del Sprint 3

Pendiente	En Desarrollo	En Testeo	Terminada
HU17	HU14		HU01
HU18	HU15		HU02
	HU16		HU03
			HU04
			HU05
			HU06
			HU07

			HU08
			HU09
			HU10
			HU11
			HU12
			HU13

3.2.3.4.1 Retrospectiva Sprint 3

Se obtuvo algunas demoras durante el desarrollo del sprint por desconocimiento del manejo de los distintos APIs, el tiempo desplazado fue remediado en la etapa de configuración de los sintetizadores.

En la ilustración 30 se pudo visualizar la existencia de tareas que tomaron más tiempo del estimado y generaron un desbalance en la distribución de trabajo a pesar de esto el sprint 3 finalizó de acuerdo a lo estimado, en tabla 22 se pudo visualizar que todos los parámetros se cumplieron a pesar mala distribución de trabajo, las tareas fueron cumplidas y se mejoró en algunos parámetros como son la comunicación y el trabajo en equipo.

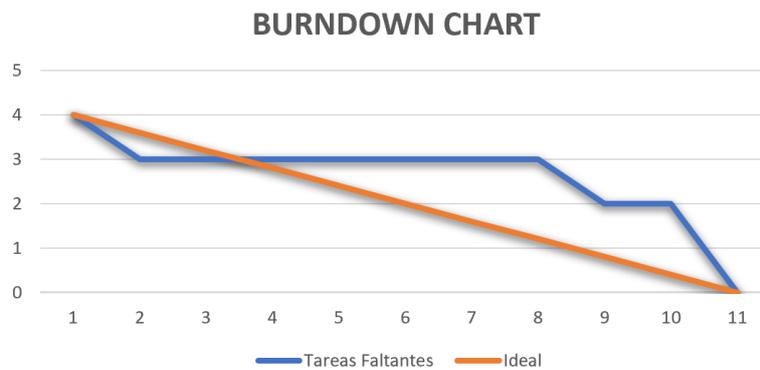


Ilustración 30 Burndown Chart Sprint 3.

Tabla 22 Retrospectiva usando controles de mejoramiento Sprint 3

Parámetro	Calificación	Semáforo	Observaciones
Gestión de cambios	10		En esta etapa el interesado no solicitó cambio alguno
Comunicación con el cliente	10		Las reuniones se realizaron los días acordados y la prueba de uso salió exitosa
Tareas cumplidas	10		Con algo de retraso se logró finalizar las actividades
Trabajo en equipo	10		Sin inconvenientes

3.2.3.5 Sprint 4

Para este sprint fue necesario definir los requerimientos para la presentación de estadísticas correspondientes al plugin desarrollado para su adaptación en WordPress y la app para ejecutarse en los kioscos. Los criterios de usuario fueron definidos en el Anexo G en las tablas HU14, HU15 y HU16, en la Tabla 23 se detalló y resumió los criterios de aceptación y las observaciones del cliente.

Tabla 23 Criterios de aceptación Sprint 4

Cod	Historia de Usuario	Criterio de aceptación	SI/NO	Observaciones
HU14	Obtener interfaz de selección de opciones	Menú de navegación	SI	
		Presentación Institucional	SI	
HU15	Presentar estadísticas de asistente móvil	Interfaz gráfica y opciones	SI	
		Presentación de cifras	SI	
HU16	Presentar estadísticas de asistente web	Interfaz gráfica y opciones	SI	
		Presentación de cifras y estadísticas	SI	

Después de la ejecución del Sprint 4, el tablero Kanban se mostraba de la siguiente manera.

Tabla 24 Kanban del sprint 4

Pendiente	En Desarrollo	En Testeo	Terminada
	HU17	HU14	HU01
	HU18	HU15	HU02
		HU16	HU03
			HU04
			HU05
			HU06
			HU07
			HU08
			HU09
			HU10
			HU11
			HU12
			HU13

3.2.3.5.1 Prototipo Sprint 4

El prototipo diseñado para este sprint contuvo:

- Menú de navegación
- Graficas estadísticas y cifras

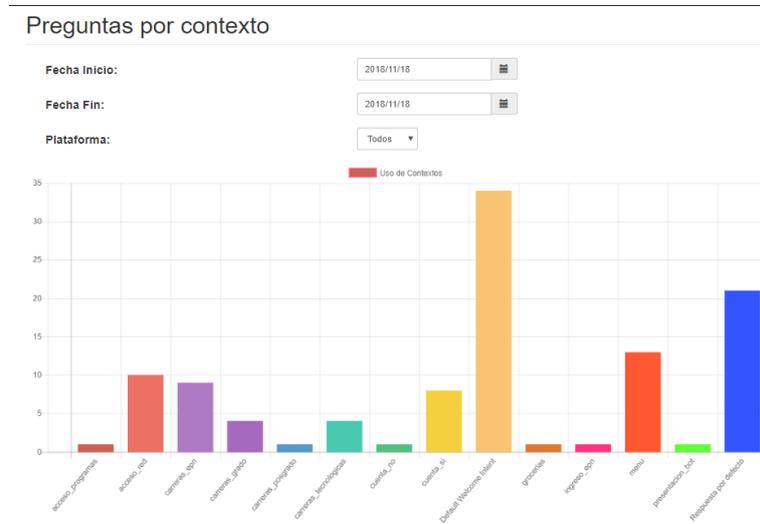


Ilustración 31 Prototipo visualización interfaz estadísticas

3.2.3.5.2 Retrospectiva Sprint 4

Los productos y prototipos se finalizaron en el tiempo establecido sin ninguna novedad.

En la ilustración 32 se pudo visualizar las tareas fueron cumplidas antes tiempo del estimado para el sprint 4, esto permitió más flexibilidad de dos días en los próximo Sprint, en tabla 25 se pudo visualizar que los parámetros evaluados cumplen con lo esperado y la sinergia del equipo va mejorando.

BURNDOWN CHART

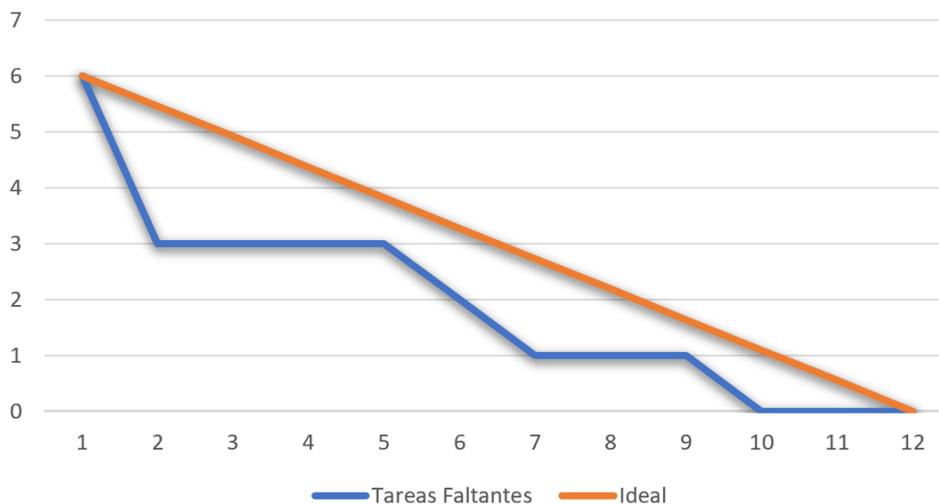


Ilustración 32 Burndown Chart Sprint 4

Tabla 25 Retrospectiva usando controles de mejoramiento Sprint 4

Parámetro	Calificación	Semáforo	Observaciones
Gestión de cambios	10		Lo acordado fue realizado sin necesidad de realizar cambios
Comunicación con el cliente	10		Las reuniones se realizaron los días acordados y la entrega del prototipo se realizó con normalidad
Tareas cumplidas	10		Se termino antes de lo estimado
Trabajo en equipo	10		El desarrollo cumplió con lo planificado

3.2.3.6 Sprint 5

Para finalizar el siguiente sprint se dedicó para la generación del formulario fa-201 solicitado por el interesado. Las historias de usuario fueron detalladas en el Anexo G en las tablas HU17 y HU18, en la Tabla 26 se detalló y resumió los criterios de aceptación y las observaciones del cliente.

Tabla 26 Criterios de aceptación Sprint 5

Cod	Historia de Usuario	Criterio de aceptación	SI/NO	Observaciones
HU17	Presentar formulario	Botón para documento	SI	
		Ventana de visualización	SI	
HU18	Generar documento formulario lleno	Generación de Documento Lleno	SI	
		Generador de documentos	SI	

Después de la ejecución del Sprint 5, el tablero Kanban se mostraba de la siguiente manera.

Tabla 27 Kanban del Sprint 5

Pendiente	En Desarrollo	En Testeo	Terminada
			HU01
			HU02
			HU03
			HU04
			HU05
			HU06
			HU07

			HU08
			HU09
			HU10
			HU11
			HU12
			HU13
			HU14
			HU15
			HU16
			HU17
			HU18

3.2.3.6.1 Prototipo Sprint 5

El prototipo diseñado para este sprint contuvo:

- Ejemplo de generación de documentos

Formulario: FAA_201

Formulario FAA 201.
Formulario para supresión de matrícula

Facultad *
Facultad de Ciencias Administrativas

Carrera *
Ingeniería de la Producción

Ingrese el periodo *
2018 B

Ingrese sus nombres *
Katherine Lizeth

Paso 1 Cancelar

Ilustración 33 Llenado de datos desde un formulario HTML

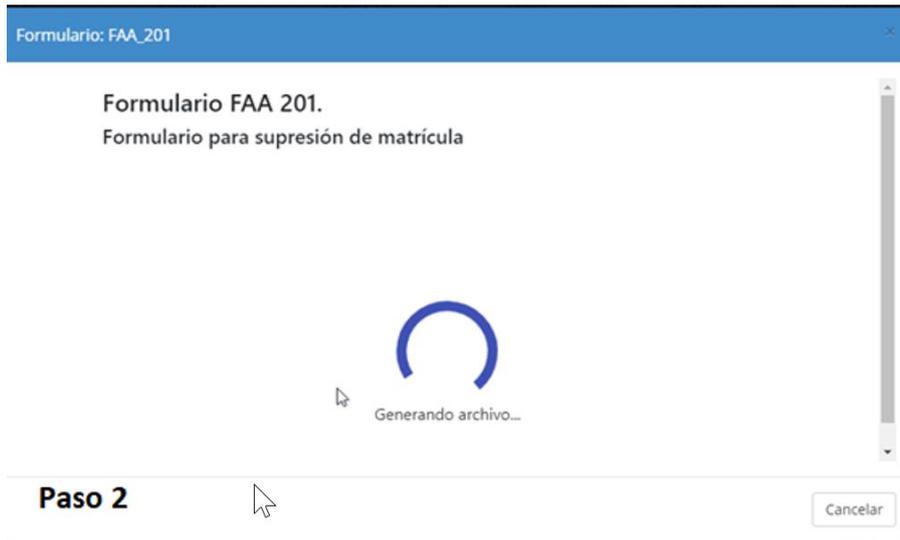


Ilustración 34. Generación del formulario

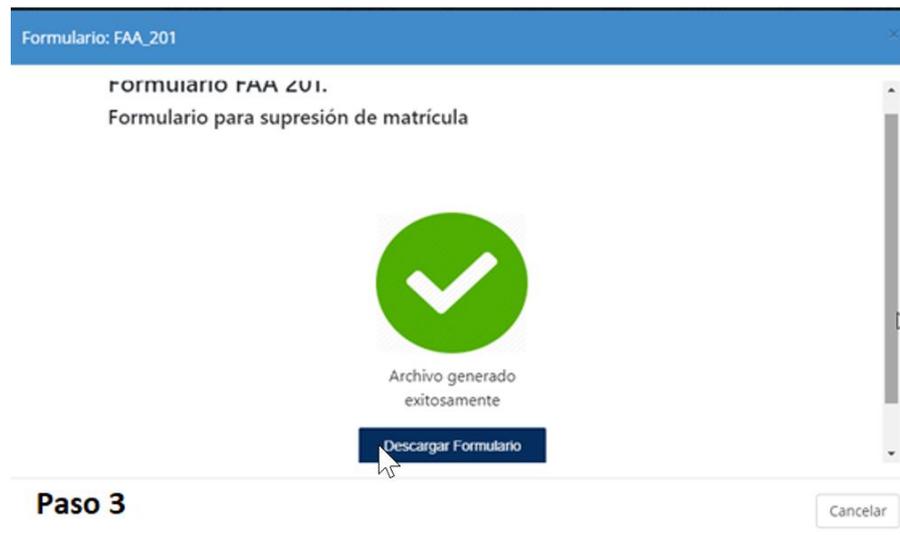


Ilustración 35. Generación exitosa del formulario

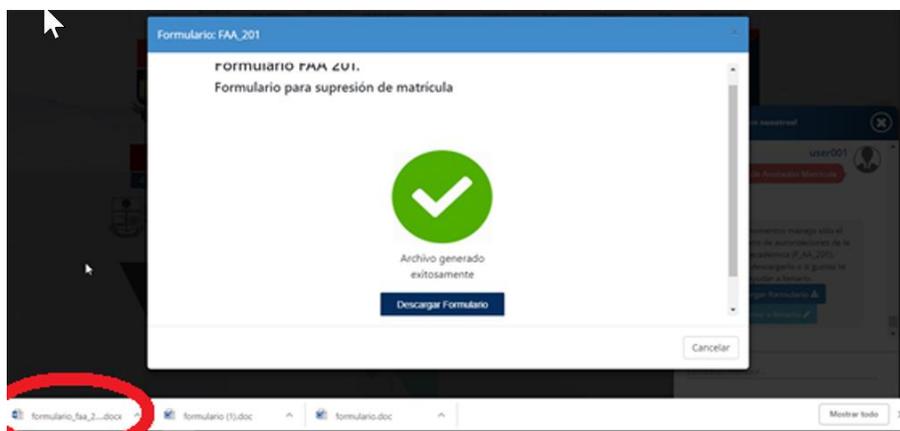


Ilustración 36. Descarga del formulario generado

formulario_faa_201_30d4f1eb97345ef9d5a05948bf3bc.docx - Vista protegida - Se ha guardado en Este PC Katherine Hurtado

ESCUELA POLITÉCNICA NACIONAL
VICERECTORADO DE DOCENCIA
Aprobación 30 junio 2009, Cambios Aprobados 11/09/2014, Cambios Aprobados 24/09/2016
UNIDAD ACADÉMICA: Facultad de Ciencias Administrativas
AUTORIZACIONES DE LA UNIDAD ACADÉMICA

F_AA_201

FECHA PEDIDO	18/09/2018
CARRERA / PROGRAMA	Ingeniería de la Producción
PERÍODO	2018 B
CÉDULA ESTUDIANTE	1724347339
NOMBRE ESTUDIANTE	Katherine Lizeth Hurtado Moína
Favor marcar con una "x" el casillero seleccionado	
ESTUDIANTE SOLICITA	
AUTORIDAD CONCEDE:	
PAGOS EN PARTES	Decano
PAGO CON CRÉDITO DEL IECE	Decano
MATRICULA EXTRAORDINARIA	Decano
MATRICULA EXTRAORDINARIA, SIN RECARGO	Decano
PRORROGA MATRICULA UNIDAD DE TITULACIÓN	Decano
CAMBIO OPCIÓN TITULACIÓN	Opción Actual Nueva Opción Decano
SUPRESIÓN DE REGISTRO DE MATRÍCULA	X Subdecano
ANULACIÓN DE MATRICULA EN MÓDULOS	Subdecano
CÓDIGO MÓDULO	NOMBRE MÓDULO

AUTORIZACION

Ilustración 37. Documento generado

3.2.3.6.2 Retrospectiva Sprint 5

Se obtuvo algunas demoras durante el desarrollo del sprint por desconocimiento del manejo del software y la creación del motor java necesario para la generación del documento, se cumple con el tiempo establecido gracias al uso de un API de ayuda en la codificación del motor JAVA

En la ilustración 38 se pudo visualizar la existencia de tareas que tomaron más tiempo del estimado a pesar de esto el sprint 5 finalizó de acuerdo a lo estimado, en tabla 28 se pudo visualizar que uno de los parámetros con más inconvenientes fue trabajo en equipo esto debido al manejo de tiempo y designación de tareas.

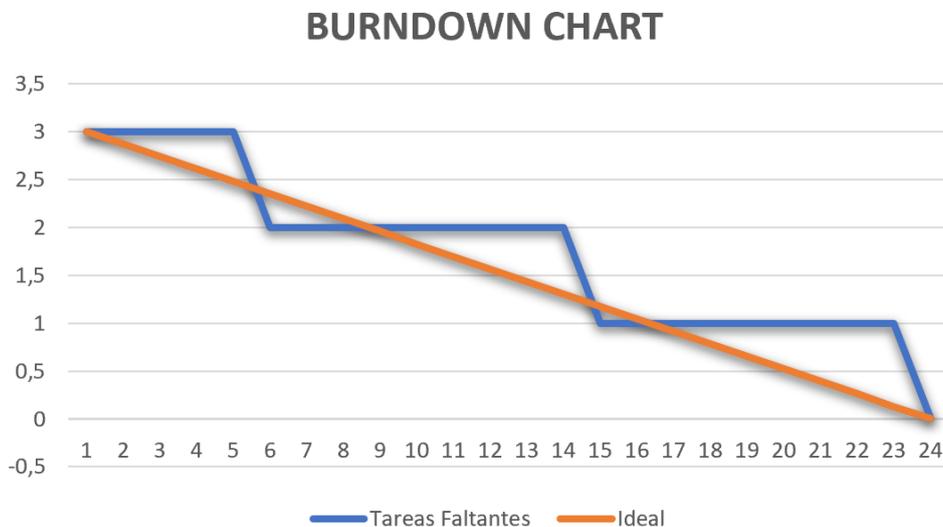


Ilustración 38. Burndown Chart Sprint 5

Tabla 28. Retrospectiva usando controles de mejoramiento Sprint 5.

Parámetro	Calificación	Semáforo	Observaciones
Gestión de cambios	10		Sin solicitud de cambios por parte del interesado
Comunicación con el cliente	10		Las reuniones se realizaron los días acordados y la entrega del prototipo se realizó con normalidad
Tareas cumplidas	10		Con algo de retraso se logró finalizar las actividades
Trabajo en equipo	5		falta de comunicación y de herramientas colaborativas

3.2.4 Estimación de costos de desarrollo

Se determino un costo estimado necesario para el desarrollo del sistema sin tener en cuenta costos asociados al uso del servidor, su mantenimiento y los dispositivos como son tablets. Los valores estimados para talento humano fueron calculados en base a costos por hora promedio pagados en el mercado de desarrollo de software dentro del país. Para el salario del Product Owner fue necesario tomar en cuenta el número de horas en la que este realiza su trabajo en el desarrollo del proyecto.

Tabla 29. Costos relacionados al talento humano

Perfil	Costo por Hora (USD)	Horas de trabajo	Personas	Total USD
Product Owner	14,50	40	1	580,00
Scrum Master	11,00	30	1	330,00
Desarrolladores	7,00	240	2	1.680,00
			TOTAL	2.590,00

Tabla 30. Estimación de costos no tecnológicos

Descripción	Costo (USD)	Meses	Total (USD)	
Licencia API VOZ(Propietaria)	35	2	70,00	
Licencia Publicación App	39,99	Única	39,99	
Internet	27,00	2	54,00	
Luz	30,00	2	60,00	
			TOTAL	223,99

El costo aproximado para el desarrollo de este proyecto fue estimado en 2.000,00 dólares americanos.

3.3 Arquitectura usada dentro del proyecto

Se tomo en cuenta la arquitectura definida por el API de DialogFlow expresada en la ilustración 39, la cual muestra una visión simplificada del uso del aplicativo y sus módulos de interacción.

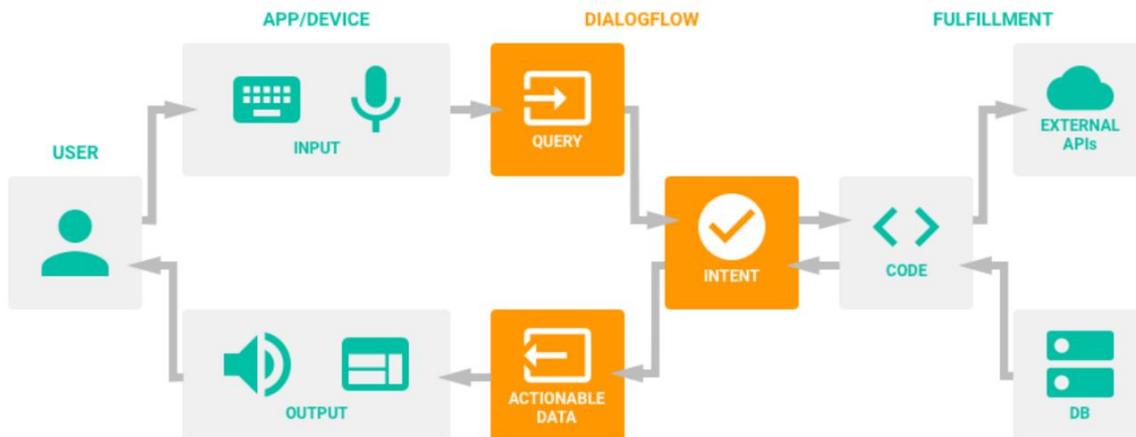


Ilustración 39. Visión simplificada de DialogFlow

Fuente:[46]

Los módulos se dividen en 3:

- Módulo de comunicación
 - Este módulo está encargado de comunicación entre el usuario y el API
 - Este módulo maneja el ingreso y el egreso de datos por texto y voz
- Módulo del API DialogFlow
 - En este módulo se maneja las preguntas y respuestas obtenidas del API
 - En el módulo del API se añade la multimedia para las respuestas del usuario.
- Módulo de Conocimiento
 - El módulo de conocimiento es fundamental por los datos guardados en el servidor de DialogFlow
 - El conocimiento generado por el módulo de conocimiento puede venir de fuentes externas o de una base de datos.

3.3.1 Modelo Lógico

El proyecto fue desarrollado bajo una arquitectura del tipo cliente-servidor donde los clientes están alojados en dispositivos móviles o en un navegador web interactuando con el servidor por medio de peticiones REST. La ilustración 40, describe los elementos de la solución.

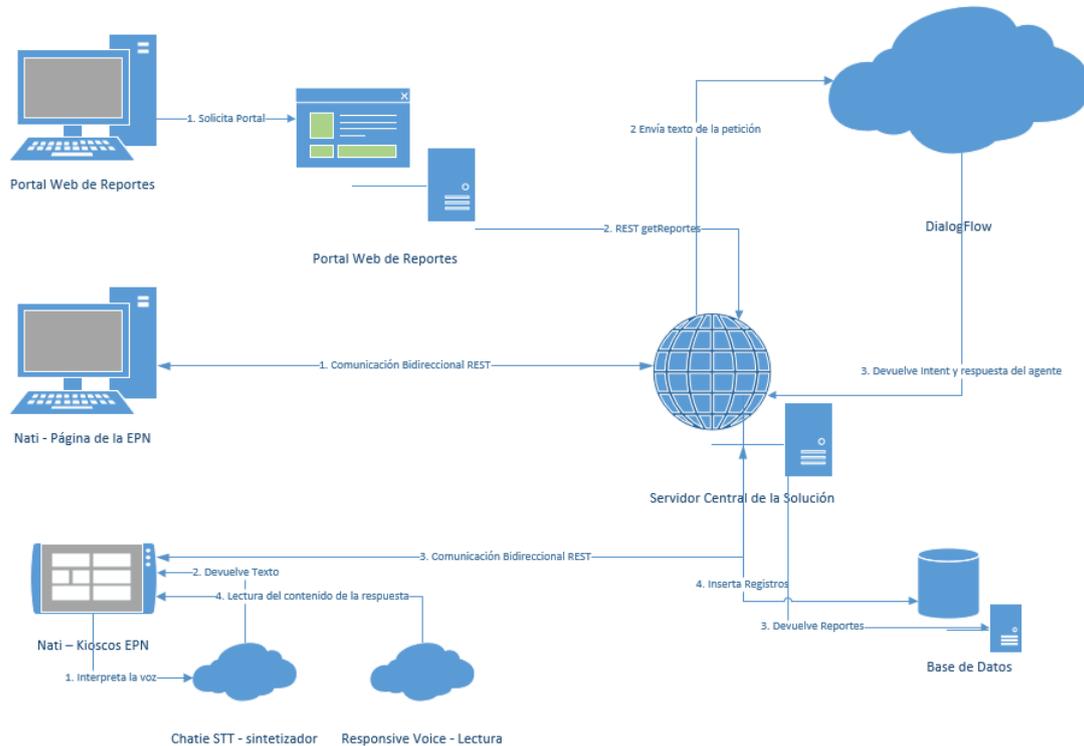


Ilustración 40. Arquitectura de la solución

La arquitectura de la solución mostrada en la ilustración 15 cuenta con 4 partes fundamentales:

- El portal web de reportes
 - Encargado de presentar los reportes en un aplicativo web haciendo uso de REST para obtener los datos.
 - Este módulo es exclusivo para los administradores de los aplicativos.
- Nati – Plugin Pagina EPN
 - Esta establece una comunicación bidireccional con el API de DialogFlow para generar la comunicación con el usuario.
 - Este módulo es el encargado de la comunicación entre el plugin desarrollado para WordPress y el API
- Nati – Kioscos EPN
 - El módulo del aplicativo instalado en los kioscos es usado para generar la comunicación entre la aplicación móvil con tecnología web y el API de DialogFlow, adicionalmente este módulo se encarga de la conversión de voz a texto y viceversa.
- DialogFlow-Applicativo
 - Este módulo contiene las configuraciones y codificación necesaria para el uso de los servicios de DialogFlow.

3.3.1.1 *Front End*

1. **Nati – Kioscos de la EPN:** la solución para los kioscos fue desarrollada en una aplicación híbrida que es capaz de ser ejecutada en los dispositivos que actualmente funcionan como kioscos.

Las herramientas utilizadas para este desarrollo se detallan a continuación:

- a. Chatie STT: API para interpretación de voz
 - b. Responsive Voice: API para lectura de textos devueltos por el servidor
 - c. jQuery, CSS3 y HTML 5 para el web view donde se mostrará el asistente virtual.
 - d. La aplicación nativa es desarrollada en Android, compatible con las tablets que funcionan como kioscos.
2. **Nati – Página de la EPN:** la solución para la página web de la EPN fue desarrollada a través de un plugin compatible con WordPress ya que la página está desarrollada sobre este gestor de contenido, por ende, las configuraciones e instalación del plugin se realizaron en PHP. La gráfica del chat y la lógica se desarrollaron con jQuery, HTML 5 y CSS3
 3. **Portal de reportes:** El portal es desarrollado en JavaScript con una API que permita graficar los registros de acuerdo con el requerimiento definido por el cliente.
 4. **Formulario F_AA_201:** Este formulario ha sido desarrollado con Angular V6 (framework de desarrollo WEB)

3.3.1.2 *Backend*

1. **Base de Datos:** Con el fin de mantener un registro de actividad de los usuarios en las aplicaciones se almacenan las actividades en la base de datos para luego obtener los respectivos reportes. Como gestor de base de datos se utiliza MySQL.
2. **Servidor de web:** El servidor fue desarrollado en NodeJS, que es un entorno que permite la implementación de servidores en código JavaScript V8 desarrollado por Chrome, este framework permite un manejo de peticiones bastante alto en comparación con un servidor implementado en JAVA que para atender peticiones en paralelo implementa hilos los cuales ocupan recursos de memoria del equipo en el que se ejecutan.

Gracias a su framework, Node JS compila el código JavaScript a un lenguaje más sencillo y liviano que la máquina podrá ejecutar sin tener que interpretar las instrucciones antes.

Node JS al manejar un sistema de eventos de entrada y salida no bloqueantes hace que sea más eficiente y liviano, se pueden atender varias peticiones en paralelo sobre eventos, Node JS tiene un ciclo de eventos de subproceso único que se detallará más adelante.

Una de las ventajas de utilizar Node JS es su amplia gama de librerías NPM disponibles para facilitar el desarrollo del servidor, entre ellas se tiene express para manejo de peticiones HTTP, para el almacenamiento y servicio de archivos estáticos, apiai (ahora DialogFlow) para la integración con la API del asistente virtual entre otras. La mayoría de las plataformas, APIS de PLN Y Machine Learning ofrece la posibilidad de integrarse con NodeJS como backend.

Para manejar las peticiones desde el cliente, se ha utilizado la librería **Express** que facilita la interpretación de respuestas y peticiones bajo el protocolo HTTPS, permite configurar rutas a las diferentes funcionalidades del servidor, por ejemplo, la descarga de imágenes, archivos, intercambio de datos, es decir permite la implementación de una API REST para la comunicación con el cliente.

3. **Generador de Formularios:** Es un pequeño motor cuya función es generar un formulario compatible con Word a partir de datos recibidos por sockets, el formulario ha sido diseñado en Jasper y el desarrollo bajo Java 8.
4. **DialogFlow:** Es una API propia de Google que combina el procesamiento de lenguaje natural con Machine Learning lo que la convierte en una poderosa herramienta para el desarrollo de asistentes virtuales. Bajo este concepto un asistente virtual es un ente artificial capaz de interpretar y responder las peticiones provenientes de un humano.

DialogFlow permite crear agentes inteligentes a manera de asistentes virtuales capaces de identificar varios tópicos en una conversación con un ser humano, a estos tópicos se los conoce como "Intents", entonces, un agente es un ente compuesto por "Intents" o contextos con los que éste identifica el tema de la conversación.

En la plataforma un agente tiene varias partes, las que se han configurado para el desarrollo de este proyecto son: manejo de contextos, uso de webhook y entrenamiento del asistente virtual.

El uso de webhook es la comunicación entre el servidor y la API esto se hace posible por medio de peticiones REST.

El entrenamiento del asistente involucra aprobar o descartar manualmente frases que pertenezcan a cierto contexto con el fin de mejorar la clasificación durante una conversación del agente, de esta forma se lleva a cabo el training.

La ilustración 41, muestra un esquema de la estructura de una agente en DialogFlow.

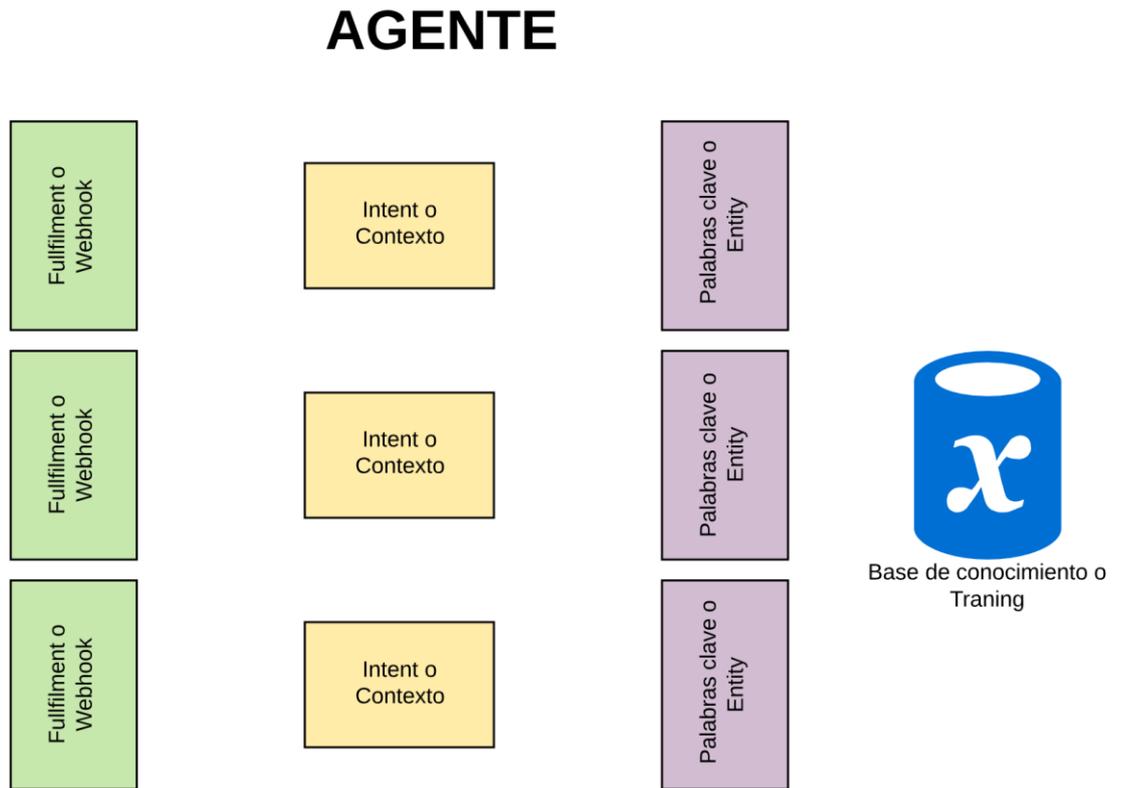


Ilustración 41. Estructura de un agente en DialogFlow

Capítulo 4 Evaluación y resultados

4.1 Instalación

4.1.1 Instalación del servidor y plugin web

La integración del plugin a la página web dependió del levantamiento del servicio de respuesta integrado con DialogFlow como fuente de conocimiento (Ver Anexo B).

4.1.1.1 Recursos Hardware

En la tabla 31,32 y 33, se muestra las distintas configuraciones de hardware correspondientes a las características del servidor y cliente a ser usadas en el proyecto.

4.1.1.1.1 Servidores

Tabla 31. Configuración de hardware del servidor

Servidor 1		
Dato	Valor mínimo	Valor recomendado
Procesador	2 núcleos	3 núcleos
Memoria RAM	8 GB	16 GB
Tamaño Almacenamiento	500 GB	1 TB

4.1.1.1.2 Estaciones cliente

Plataforma WEB

Tabla 32. Configuración de hardware del cliente web

Dato	Valor mínimo	Valor recomendado
Procesador	1 núcleo	2 núcleos
Memoria RAM	2 GB	4 GB
Tamaño Almacenamiento	500 GB	500 GB
Otros	Navegador Web (Opera, Mozilla, Chrome)	Navegador Web (Opera, Mozilla, Chrome)

4.1.1.1.3 Conectividad

Tabla 33. Configuración de hardware-conectividad

Dato	Valor mínimo	Valor recomendado
Tarjeta de Red	100 Mbps	1000 Mbps
Tipo de Red	Wifi	Cableada
Otros	N/A	N/A

4.1.1.2 Restricciones

En la tabla 34, se muestra distintas restricciones que podrían afectar al correcto desarrollo del sistema.

Tabla 34. Restricciones físicas del proyecto

Restricción	Detalle
Baja velocidad de conexión a Internet	Al funcionar con APIS de terceros, si la solución no se puede comunicar con estas ninguno de los asistentes virtuales funcionará de acuerdo con los requerimientos
Conexión a internet en el servidor	Si el servidor no tiene conexión a internet, los asistentes virtuales no funcionarán.
Permisos de ejecución y Publicación	El dominio smartbot.epn.edu.ec debe estar publicado de lo contrario no podrá ser invocado desde el exterior.

4.1.1.2.1 Restricciones técnicas del sistema

En la tabla 35, se muestra las configuraciones correspondientes al sistema operativo presente en el servidor donde se ejecuta la solución del proyecto.

Tabla 35. Restricciones técnicas del sistema

Elemento	Descripción
Sistema operativo	CentOS 7.0
Servidor de aplicaciones	Node JS v6.14.3
Servidor de base de datos	MySQL Ver 14.14 Distrib 5.7.23, for Linux (x86_64) using EditLine wrapper
Compilador	Gcc
JVM	OpenJDK 1.8.0_181
Otros	

4.1.1.3 INSTALACIÓN Y CONFIGURACIÓN DEL SOFTWARE BASE

Para el funcionamiento del servidor fue necesario de los siguientes paquetes, explicados en la tabla 36, 37 y 38.

Tabla 36. Instalación NodeJS

NODE JS	
Descripción	Software para levantamiento del servidor
Localización	/bin/node

Procedimiento de instalación	
Paso 1	Acceder como super usuario (sudo su)
Paso 2	Instalar compilador de C con el comando yum install -y gcc-c++ make
Paso 3	Descargar la version de Node con el comando: curl -sL https://rpm.nodesource.com/setup_6.x sudo -E bash -
Paso 4	Instalar Node JS con el comando: yum install nodejs
Paso 5	Revisar la versión de NODE con el comando: node -v y la versión de NPM con el comando npm -v
Procedimiento de configuración	
Paso 1	Generar solicitud de certificados
Paso 2	Almacenar certificados originados en la Entidad Certificador en la carpeta donde se vaya a configurar el servidor
Paso 3	Instalar forever con el comando: npm install forever --save

Tabla 37. instalación paquete java

JAVA	
Descripción	Software para manejar el generador de formularios
Localización	/bin/java
Procedimiento de instalación	
Paso 1	Ejecutar el siguiente comando: sudo yum install java-1.8.0-openjdk-devel
Paso 2	Revisar la versión instalada de JAVA con el comando: java -version
Procedimiento de configuración	
Paso 1	Obtener las fuentes de Microsoft para generar los formularios con el comando: wget http://nchc.dl.sourceforge.net/project/mscorefonts2/rpms/msttcore-fonts-installer-2.6-1.noarch.rpm
Paso 2	sudo rpm -ivh msttcore-fonts-installer-2.6-1.noarch.rpm

Tabla 38. instalación MYSQL

MYSQL	
Descripción	Gestor de Base de Datos
Localización	/usr/bin/mysql
Procedimiento de instalación	
Paso 1	Descargar el paquete con el siguiente comando: wget https://dev.mysql.com/get/mysql57-community-release-el7-9.noarch.rpm
Paso 2	Ejecutar el siguiente comando md5sum mysql57-community-release-el7-9.noarch.rpm

Paso 3	Instalar el paquete descargado con el comando: sudo rpm -ivh mysql57-community-release-el7-9.noarch.rpm
Paso 4	Instalar el servidor de base de datos con el comando: sudo yum install mysql-server
Paso 5	Iniciar el servicio con el comando: sudo systemctl start mysqld
Paso 6	Verificar el estado del servicio con el comando: sudo systemctl status mysqld
Procedimiento de configuración	
Paso 1	Cambiar la clave temporal de acceso con el comando: sudo grep 'temporary password' /var/log/mysqld.log Aparecerá la clave temporal que se debe cambiar
Paso 2	Cambiar la clave temporal con el siguiente comando: sudo mysql_secure_installation
Paso 3	Para acceder a la instalación segura utilice la clave definida en el paso 1

Express

- Express es una infraestructura de aplicación web NodeJS
- Express es una API con métodos de programa de utilidad HTTP y middleware.
- Express permite el manejo de rutas de servidor
- Express permite el manejo y configuración central del servidor
- Express permite servir aplicaciones y archivos estáticos por medio del protocolo HTTP o HTTPS

En la ilustración 42, se muestra el código necesario para iniciar los servicios necesarios para la ejecución del agente conversacional.

```

package.json x
1  {
2    "name": "botsfactory",
3    "version": "1.0.0",
4    "description": "Servidor de aplicaciones",
5    "main": "index.js",
6    "scripts": {
7      "test": "node",
8      "start": "node ./bin/www"
9    },
10   "author": "Katherine Hurtado - Jonathan Zúñiga",
11   "license": "ISC"
12  }

```

Archivo de inicio del server

Ilustración 42. Código para el inicio automático del servidor

El archivo `./bin/www` contiene las configuraciones del servidor como son:

- Certificados
- Puerto del servidor
- Levantamiento del servidor
- Monitoreo de errores

El archivo **app.js** tiene las configuraciones de express como son:

- Configuraciones de la librería
- Configuración de rutas
- Uso de otros módulos

En la ilustración 43, se muestra la codificación necesaria para hacer uso del webhook del asistente.

```
const fs = require('fs');
const express = require('express');
const bodyParser = require('body-parser');
const app = express();
const verificationController = require('./controllers/verification');
const messageWebhookController = require('./controllers/messageWebhook');
const reportesWebhookController = require('./controllers/reportesWebhook');
var log4js = require('log4js');
//Log de información para errores de programación
var log = log4js.getLogger("app");
var favicon = require('static-favicon');
app.use(favicon());

app.use(bodyParser.json());
app.use(bodyParser.urlencoded({ extended: true }));
app.use(express.static(__dirname + '/private'));
app.use(log4js.connectLogger(log4js.getLogger("http"), { level: 'auto' }));
```

Ilustración 43. Codificación Webhook asistente

Para poder enviar el archivo generado por el motor JAVA fue necesario asociar el archivo a la petición hecha por el usuario, en la ilustración 44, se puede visualizar los comandos necesarios para la asociación y que hacer en caso de fallo.

```
app.get('/formulario', function(req, res, next) {
  res.sendFile('./private/f_aa_201.doc', function(err) {
    if (err) {
      next(err);
      log.error("ERROR al enviar archivo de login: "+err);
    } else {
      log.info("Archivo enviado exitosamente");
    }
  });
});
```

Ilustración 44. Codificación generación de formulario

Para la apertura del módulo de reportes este hace uso de módulos importados (WebhookController)

```
//RUTA PARA REPORTES
app.get('/reportes', reportesWebhookController);
//RUTA API REST
app.post('/', messageWebhookController);
```

Ilustración 45. Codificación Webhook reportes

Para el manejo de módulos se recibe e identifica las peticiones para ser enrutadas a los módulos que procesan la información.

```

module.exports = (req, res) => {
  //Añadir un type en la petición al servidor
  if (req.body.PARAM){
    const prueba=require('uricode')(req.body.PARAM); // Decodifica los parámetros de entrada
    var json = JSON.stringify(eval("(" + prueba + ")")); //Evalua el JSON
    var message = JSON.parse(json); // Transforma la petición a JSON
    if(message.tipo){ // Evalúa el tipo en la petición llenado de formulario
      socketmessage= socketManager.procesarRequest(message.args, res, req);
    }else {
      epnw.procesarRequest(message, res); //Comunicación con DialogFlow
    }
  }
}

```

Ilustración 46. Codificación petición de llenado de formulario

4.1.2 Desarrollo e Instalación del aplicativo móvil

4.1.2.1 Estación cliente para la plataforma Móvil

En la tabla 39, se muestra las distintas configuraciones para la ejecución de la app en los dispositivos móviles ubicados en los distintos kioscos de la EPN

Tabla 39. configuración de hardware para la plataforma móvil

Dato	Valor mínimo	Valor recomendado
Procesador	Snapdragon 600	Snapdragon 600
Memoria RAM	2GB	4 GB
Tamaño Almacenamiento	8GB	8 GB
Sistema Operativo	Android 6.0	Android 6.0
Otros	Habilitado micrófono y parlante	Habilitado micrófono y parlante
Aplicaciones	Chrome Actualizado	Última versión de Chrome

4.1.2.2 Desarrollo

El desarrollo del aplicativo móvil fue dividido en 4 partes (Ver Anexo C):

- Construcción Interfaz web (uso de HTML y CSS)**
 Debido a los requerimientos necesarios para la integración del aplicativo fue necesario enfocar el desarrollo a un aplicativo móvil con tecnología web y que también este se pueda visualizar en un navegador web que mantenga habilitados los recursos de sonido como el micrófono y el parlante. La ilustración 47, muestra un ejemplo del IDE de desarrollo y una pantalla con la interfaz

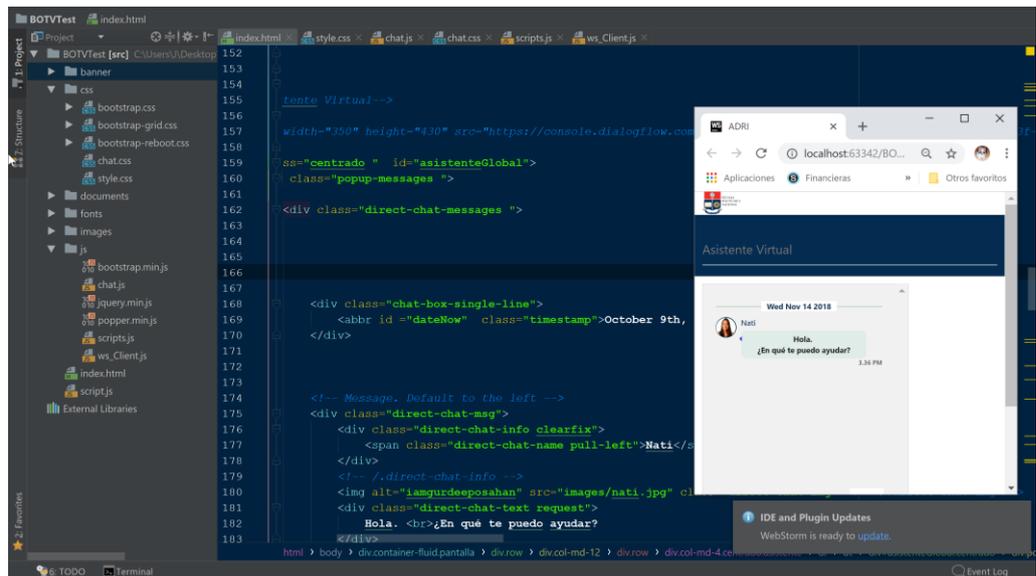


Ilustración 47. Construcción interfaz del aplicativo para los kioscos

- **Desarrollo de comunicación para la generación de respuestas a las inquietudes del usuario.**
 - **Uso de DialogFlow como fuente de conocimiento**
DialogFlow es la fuente principal de conocimiento de los chatbots por esta razón debe generar contextos y sus debidas respuestas por defecto, pero se debe tomar en cuenta que mientras transcurre el tiempo el algoritmo de aprendizaje entra en juego y el Bot aprende nuevas respuestas a los distintos contextos definidos. En la ilustración 48, se listan los distintos contextos que va a manejar el chatbot. En la ilustración 49, se muestra una configuración para el intent carreras de la EPN.

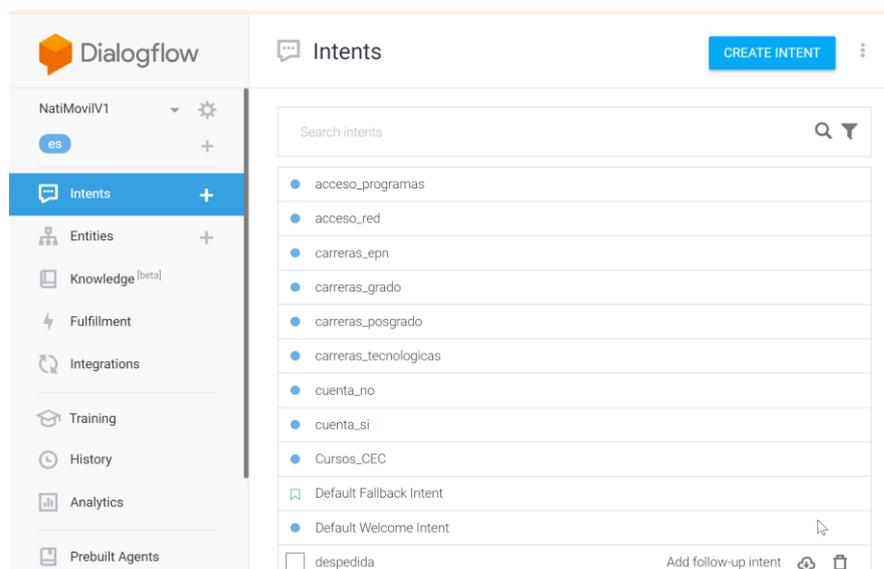


Ilustración 48. Configuración de los intents de DialogFlow

intent priority by pressing this icon

carreras_epn SAVE

” carreras epn
” ¿Qué carreras tiene la EPN?
” que carreras tiene la Poli
” Carreras de la EPN
” cuales son las carreras de la poli
” Cuál es la oferta académica de la EPN
” oferta académica de la Poli
” carreras poli
” cuál es la oferta académica de la Poli

Ilustración 49. Ejemplo de intents carreras de la EPN

- Entrenamiento previo

El Bot cuenta con distintos intents, pero para este caso se tomará en cuenta el contexto carreras de la EPN como ejemplo este debe ser revisado periódicamente en la sección de entrenamiento como se muestra en la ilustración 50 para que su base de conocimiento no se vea afectada por contenido basura

Training UPLOAD

ayuda	1	0	Oct 19	>
carreras de posgrado	1	0	Oct 19	>
carreras de grado	1	0	Oct 19	>
Carreras EPN	1	0	Oct 19	>
Formulario de Anulación Matrícula	1	0	Oct 19	>
ayuda	1	0	Oct 19	>
carreras de grado	1	0	Oct 19	>
Carreras EPN	1	0	Oct 19	>
ayuda	1	0	Oct 19	>
si tengo correo institucional	1	0	Oct 19	>
Redes Wifi	1	0	Oct 19	>

Ilustración 50. Training dentro de DialogFlow

- **Transformación voz a texto**

- **Uso de api chattie.js**

Chattie.js cuenta con un entorno de prueba como se muestra en ilustración 51, donde se pudo hacer uso de su tecnología en captación de voz y transformación de texto ya que este cuenta con varios idiomas para sus distintas pruebas.



Ilustración 51. Interfaz del api Chatti

- **Uso de obtención de texto mediante HTML5**

Actualmente HTML5 cuenta con muchas herramientas multimedia como son la captación de voz mediante el uso de distintos micrófonos, HTML5 cuenta con adaptaciones a distintas APIs que funcionan perfectamente con el navegador. En la ilustración 52 y 53, se muestra interfaz y el código que fue necesario para poder integrar un reconocedor de voz en cualquier aplicativo.



Ilustración 52. Transformador de voz a texto en HTML5

```

<input id="text">
<button onclick="talk()">Talk It!</button>
<button onclick="listen()">Voice</button>
<script src="../../bower_components/platform/platform.js"></script>
<script src="../../src/webspeech.js"></script>
<script>
  var speaker = new webspeech.Speaker();
  var listener = new webspeech.Listener();
  function talk() {
    speaker.speak("en", document.getElementById("text").value);
  }

  function listen() {
    listener.listen("en", function(text) {
      document.getElementById("text").value = text;
    });
  }
</script>

```

Ilustración 53. Codificación captadora de voz

- **Transformación texto a voz**

- Uso de api responsiveVoice.js (privativa opcional)
 ResponsiveVoive es un API gratuito que tiene la característica de permitir emitir un sintetización de voz en distintos idiomas y velocidades. En la ilustración 54, se muestra una interfaz genérica del uso del api.

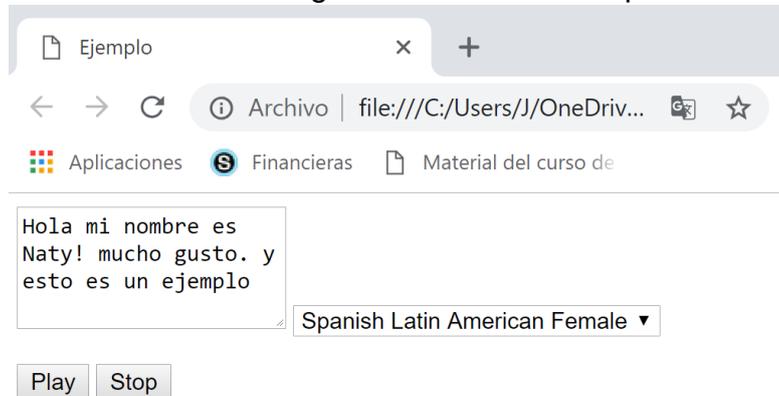


Ilustración 54. Uso de responsiveVoice.js

4.1.2.3 Integración a los dispositivos móviles

Para la integración del aplicativo al dispositivo móvil fue necesario hacer uso de Android Studio junto con la tecnología de webview la cual permite importar la aplicación con tecnología a un aplicativo móvil (Ver Anexo C)

En la ilustración 55, se puede visualizar la integración de un navegador a una aplicación móvil dentro de Android Studio.

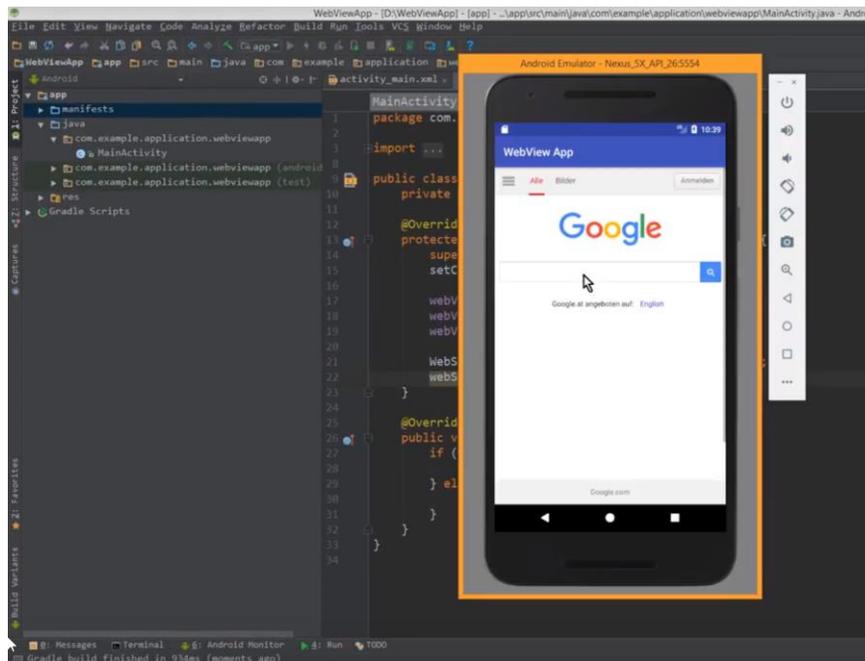


Ilustración 55. Integración de webview a un app móvil

4.2 Demostración de uso de los aplicativos

4.2.1 Aplicativo Web

El plugin está montado sobre la página WordPress de la escuela politécnica nacional. Y su icono se encuentra ubicado en la esquina inferior derecha como se muestra en la ilustración 56.



Ilustración 56. Ubicación del Chatbot

4.2.1.1 Uso de diálogos

El ChatBot cuenta con ventanas con los contextos definidos, estos contextos cuentan con el saludo y un menú con las distintas acciones de uso del SmartBot, estos contextos se encuentran de manera estática y su modificación dependerá de la necesidad específica de uso.

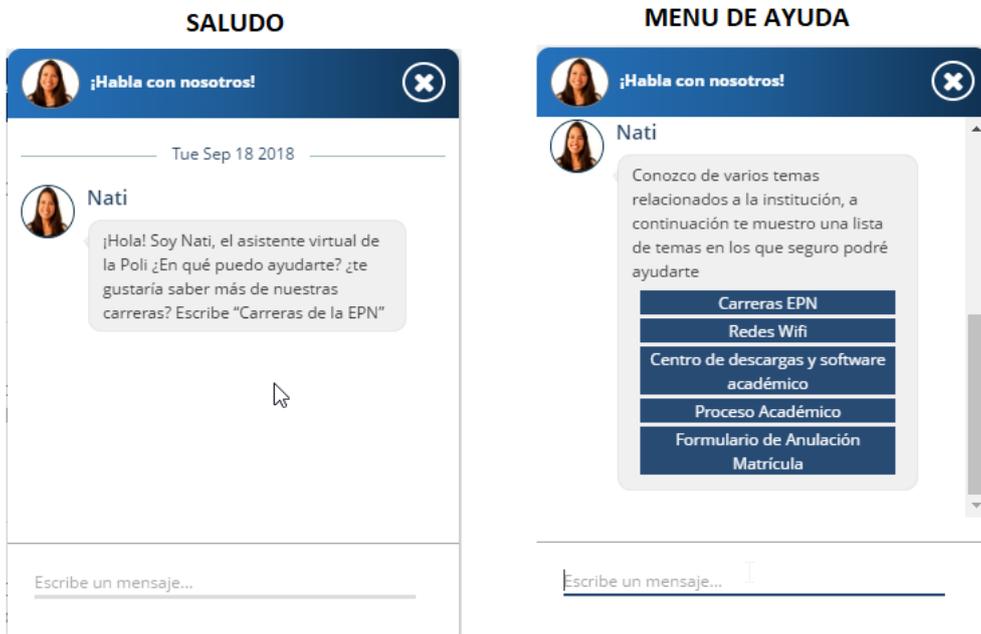


Ilustración 57. Contextos estáticos del Chatbot

Manejo del contexto “Carreras de la EPN”, este cuenta con un esquema de botones que nos permiten visualizar las distintas carreras dentro de la institución. En las ilustraciones 58 y 59, se muestran las distintas carreras por tipo de instrucción, para presentar las carreras de una manera más amigable se hizo uso de carruseles.



Carrera de Posgrado y Tecnologías



Ilustración 59. Contexto carreras de posgrado y tecnologías

El contexto “Redes Wifi”, nos permite navegar respondiendo la pregunta inicial “usted cuenta con correo institucional”, se despliega los distintos manuales dependiendo del caso, como se muestra en las ilustraciones 60, 61 y 62.

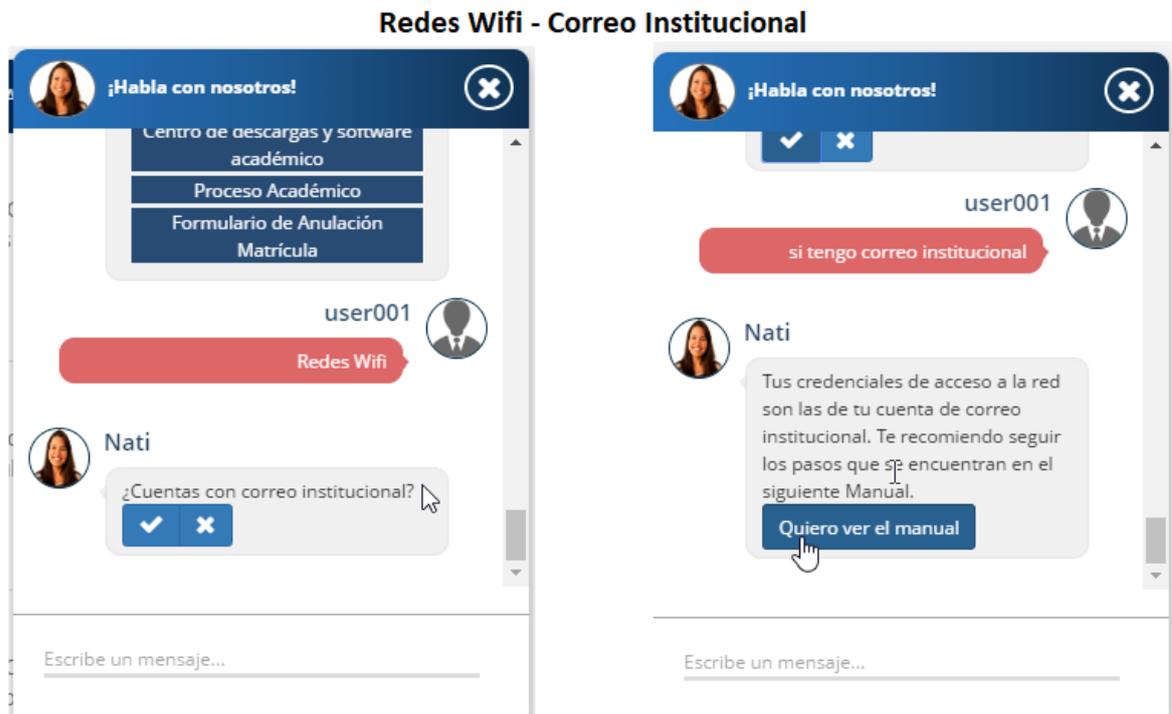


Ilustración 60. Contexto redes wifi usando correo institucional

Manual de conexión a Internet

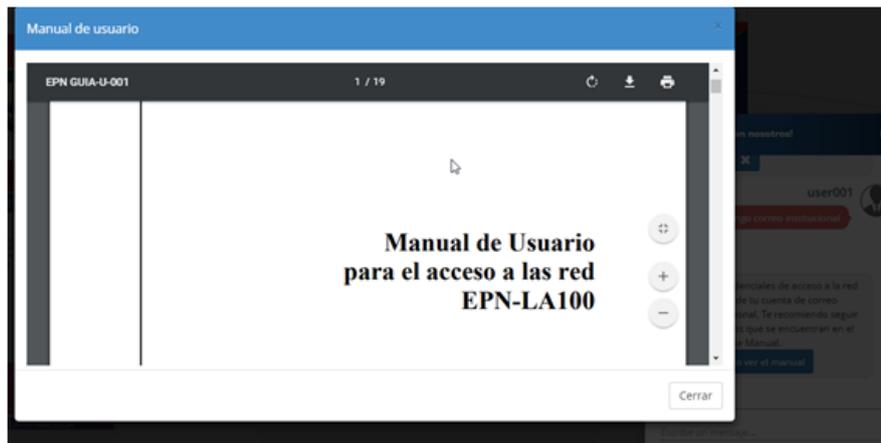


Ilustración 61. Presentación del manual de conexión a internet



Ilustración 62. contexto conexión a internet sin correo institucional

Para el contexto “Software” se cuenta con un carrusel que muestra las distintas herramientas de software disponibles en la institución además se puede visualizar una ayuda para saber cómo obtener las distintas licencias, mostradas en las ilustraciones 63 y 64.

Descargas de Software

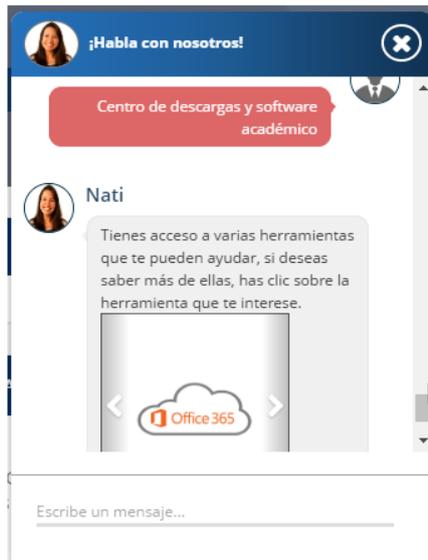


Ilustración 63. Herramientas software ofertadas por la EPN



Ilustración 64. Licenciamiento de herramientas software

Dentro del contexto “proceso académico” se cuenta con una pequeña explicación y un enlace para dirigirse a la página de admisiones, mostrada en la ilustración 65.

Proceso académico de la EPN

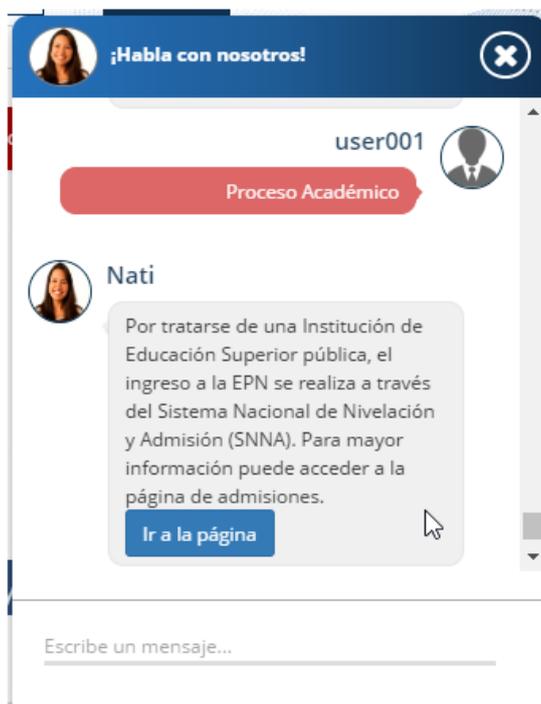


Ilustración 65. Ventana de ayuda proceso académico

Para finalizar el uso del SmartBot se cuenta con una despedida que cambia interactivamente y que sugiere otras alternativas como son el uso del tour a la EPN o un enlace para acceder al contenido de la web Page de la EPN. El SmartBot cuenta además con manejo de groserías o frases inadecuadas. Estas opciones se visualizan en las ventanas mostradas en la ilustración 66.

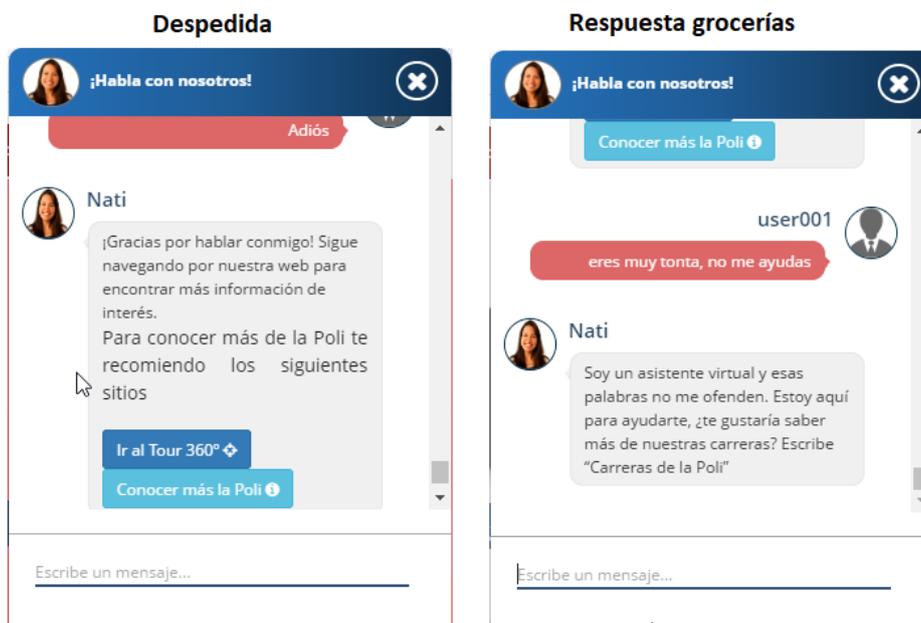


Ilustración 66. Ventana de despedida y manejo de preguntas inapropiadas

4.2.1.2 *Uso de formulario*

Para el llenado del formulario fa-201 se tiene que pulsar en el botón “ayúdame a llenarlo” o caso contrario solo será factible descargar el archivo, como se ve en la ilustración 67.

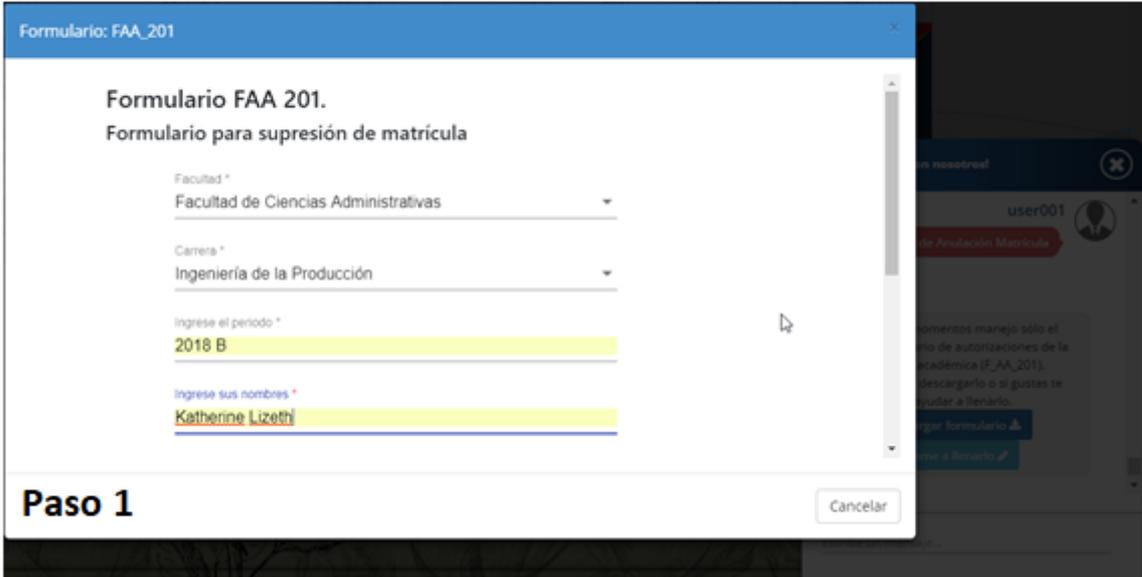
Formulario de Anulación de Matrícula



Ilustración 67. Ventana de uso de formulario

Para el llenado del formulario se puede hacer uso de un espacio HTML que permite y solicita ciertos datos para generar un archivo lleno. En las ilustraciones 68 hasta la ilustración 72, se puede visualizar cual es el proceso para genera el formulario lleno.

Llenado del Formulario



Formulario: FAA_201

Formulario FAA 201.
Formulario para supresión de matrícula

Facultad *
Facultad de Ciencias Administrativas

Carrera *
Ingeniería de la Producción

Ingrese el periodo *
2018 B

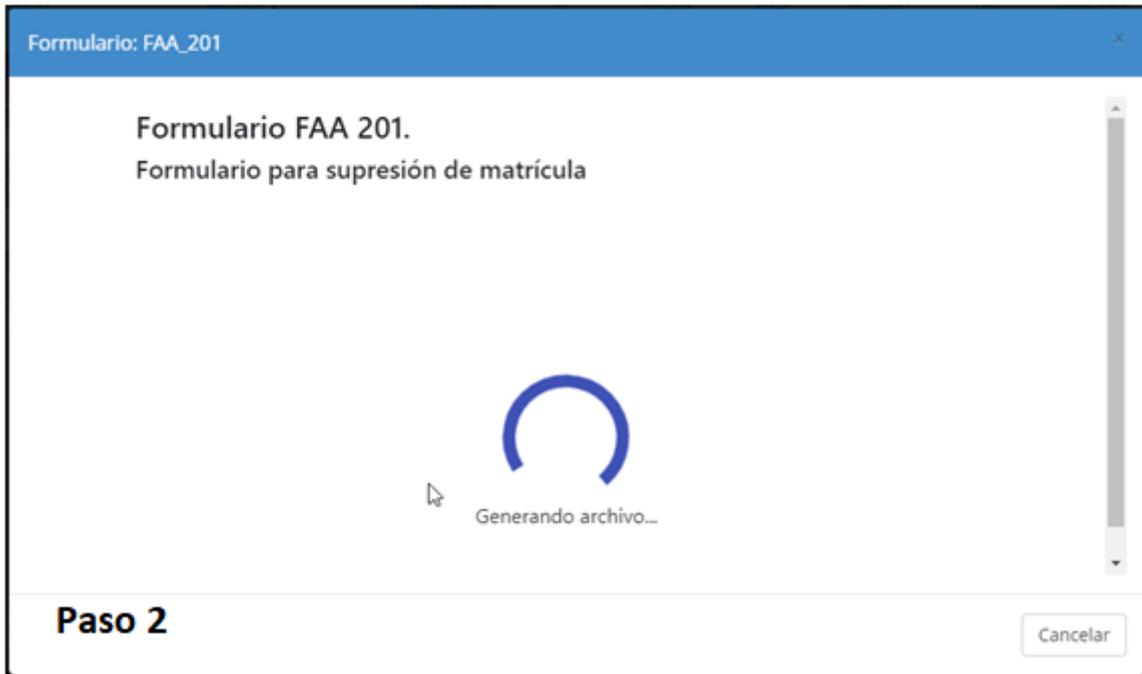
Ingrese sus nombres *
Katherine Lizethi

Paso 1

Cancelar

Ilustración 68. Ventana ayuda de llenado de formulario

Generación del formulario



Formulario: FAA_201

Formulario FAA 201.
Formulario para supresión de matrícula

Generando archivo...

Paso 2

Cancelar

Ilustración 69. Animación de generación de formulario

Formulario generado con éxito

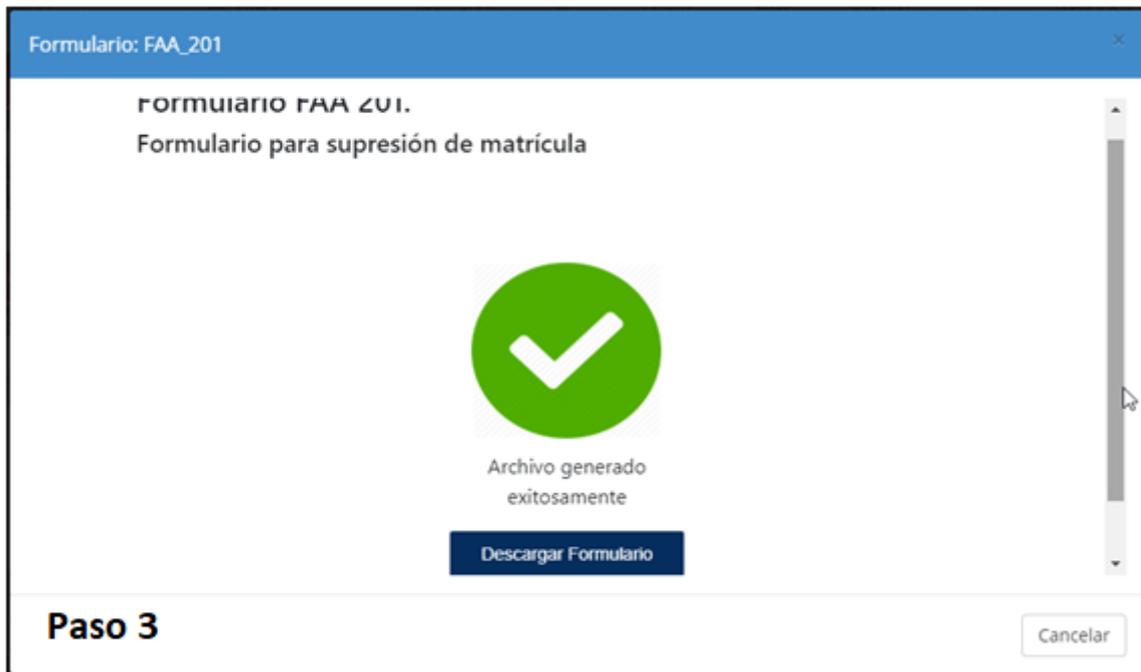


Ilustración 70. Ventana botón de descarga de formulario

Descarga del formulario

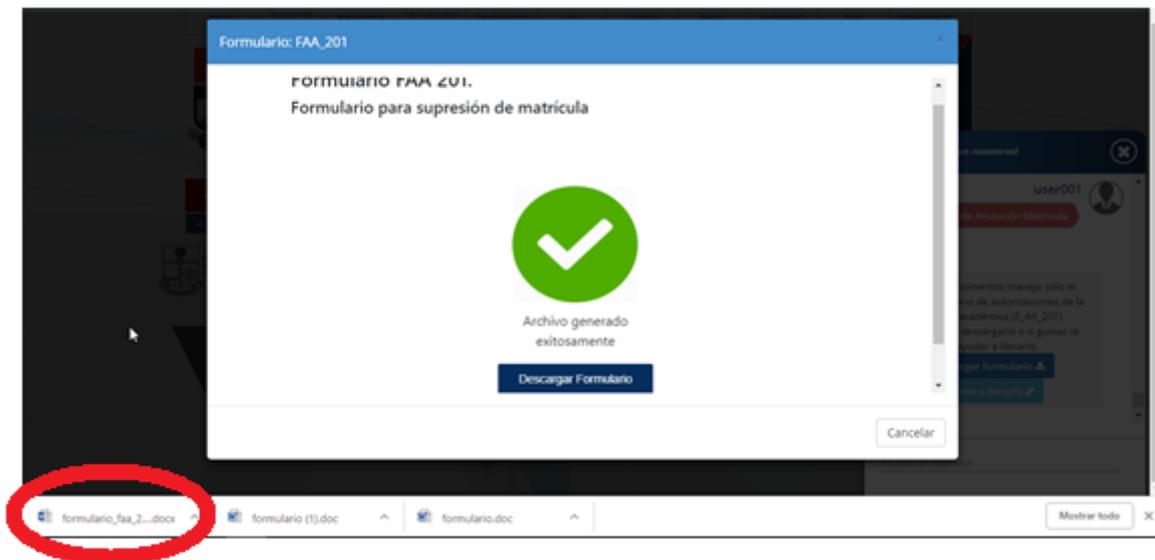


Ilustración 71. Ejemplo de enlace de documento generado

Formulario lleno con los datos ingresados

formulario_faa_201_50dd1ebc97345ef9d5a05048bf3bc.docx - Vista protegida - Se ha guardado en Este PC Katherine Hurtado

ESCUELA POLITÉCNICA NACIONAL
VICERRECTORADO DE DOCENCIA
Aprobación 30 junio 2009, Cambios Aprobados 11/09/2014, Cambios Aprobados 24/09/2016
UNIDAD ACADÉMICA: Facultad de Ciencias Administrativas
AUTORIZACIONES DE LA UNIDAD ACADÉMICA

F_AA_201

FECHA PEDIDO	18/09/2018		
CARRERA / PROGRAMA	Ingeniería de la Producción		
PERÍODO	2018 B		
CÉDULA ESTUDIANTE	1724347339		
NOMBRE ESTUDIANTE	Katherine Lizeth Hurtado Molina		
AUTORIZACIÓN	Favor marcar con una "x" el casillero seleccionado		
	ESTUDIANTE SOLICITA		AUTORIDAD CONCEDE:
	PAGOS EN PARTES		Decano
	PAGO CON CRÉDITO DEL IECE		Decano
	MATRICULA EXTRAORDINARIA		Decano
	MATRICULA EXTRAORDINARIA, SIN RECARGO		Decano
	PRORROGA MATRICULA UNIDAD DE TITULACIÓN		Decano
	CAMBIO OPCION TITULACIÓN	Opción Actual	Nueva Opción
			Decano
	SUPRESIÓN DE REGISTRO DE MATRÍCULA		X Subdecano
ANULACIÓN DE MATRICULA EN MÓDULOS		Subdecano	
CÓDIGO MÓDULO	NOMBRE MÓDULO		

Ilustración 72. Documento generado con datos llenados

4.2.1.3 Uso de Estadísticas

El acceso a la portal de generación y validación de estadísticas de los aplicativos se encuentra instalado en el servidor de producción y pruebas. Este portal muestra los datos generados por el usuario en todas las plataformas que se comuniquen con el servidor central de la solución, el aplicativo de estadísticas es de uso exclusivo para los administradores. En la ilustración 73, se muestra el uso de grafico de barras para explicar cuáles de los distintos contextos son los más usados.

Preguntas por contexto

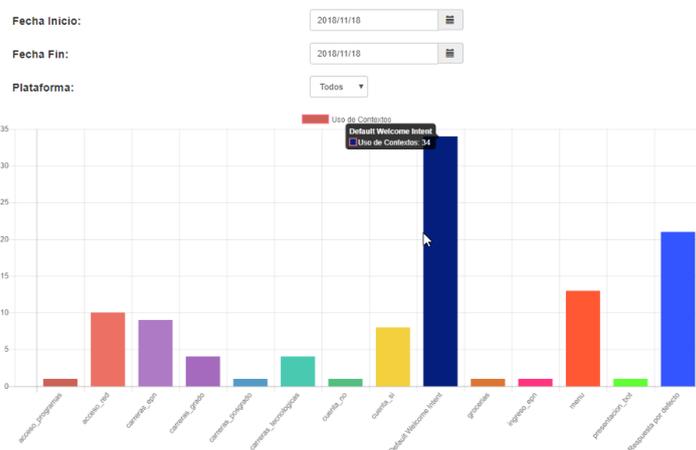


Ilustración 73. Estadísticas preguntas por contexto

En la ilustración 74, se muestra el uso de grafico de pastel para explicar el uso de las distintas plataformas durante un rango de fecha seleccionado.

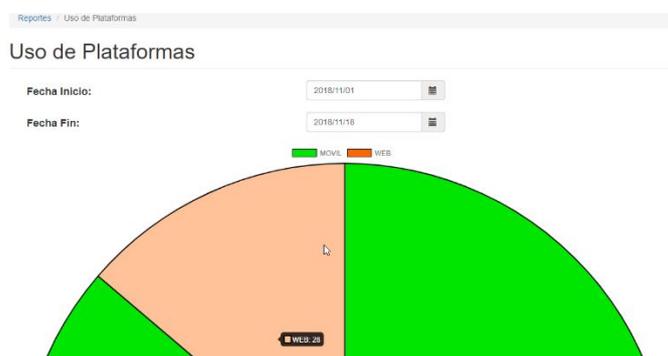


Ilustración 74. Estadísticas uso de las plataformas

4.2.2 Aplicativo Móvil con tecnología Web

El aplicativo para los kioscos cuenta con 3 campos como se muestra en la ilustración 75 y los campos son:

- Un campo de sponsor de eventos o noticias
- El asistente con funcionamiento de voz y un icono de que cambia de color en caso de comenzar a grabar
- Un documento que puede cambiar donde se carga archivos necesarios para su uso.

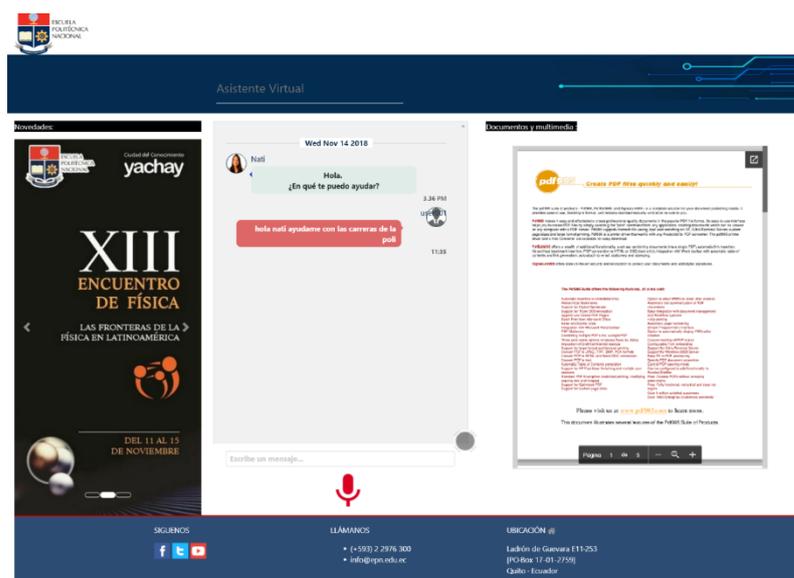


Ilustración 75. Aplicativo móvil para los kioscos

El aplicativo también es adaptativo para su ejecución en dispositivos de menor medida como se muestra en la ilustración 76, el ejemplo es la visualización en un teléfono móvil.



Ilustración 76. App móvil adaptativa

4.2.2.1 Funcionamiento TTS y STT

El funcionamiento de la captura de voz y la sintetización depende de dos APIs, la cuales son las encargadas de obtener la voz y convertirla en texto, deben ser capaces de dar una respuesta sonora que ayude al usuario con su requerimiento.

Para visualizar un caso de ejemplo del uso de la tecnología, se puede hacer uso del anexo H.

4.3 Evaluación de rendimiento

Las pruebas de carga se realizaron con la herramienta Jmeter, la cual permite: Definición de Escenarios, Ejecución de Escenarios, Análisis de resultados, además se hizo uso herramientas nativas del sistema operativo de CentOS para definir los límites de hardware con los que se contaba en el momento, estas pruebas se realizaron antes y después de cargar las diferentes tecnologías necesarias para el funcionamiento del servicio de SmartBot.

Se realizaron varias sesiones de pruebas de carga, en las que se ejecutan tres escenarios distintos, esto con el objetivo de garantizar el debido proceso para cumplir la ejecución del servicio de SmartBot sin afectar el correcto funcionamiento del servidor. Para esto fue necesario analizar el comportamiento del SmartBot NATI en los siguientes escenarios:

- **Escenario inicial del servidor:** Este escenario tomo en cuenta el levantamiento de los servicios para el funcionamiento del SmartBot. Este escenario sirvió para garantizar que durante un largo periodo de tiempo el sistema se muestra estable, y no existen consumos excesivos de CPU, memoria y conexiones.

- **Escenario de negocio:** Este escenario hizo uso de un número de usuarios concurrentes haciendo uso del api y la creación del formulario (120 usuarios por minuto aproximadamente y la creación de 100 formularios con 10 usuarios en el motor java). El objetivo de este escenario fue medir tiempos medios de respuesta, y analizar el comportamiento del sistema simulando un entorno real de consulta.
- **Escenario de saturación:** Este escenario conto con un exceso de peticiones, este exceso de petición tuvo como objetivo alcanzar el umbral en el cual el sistema dejara de responder correctamente, en este punto se visualizó los valores incrementales en el uso del procesador y trabajo del disco duro.

Después de la ejecución de los escenarios descritos y el posterior análisis de cada uno de ellos se obtuvo las siguientes conclusiones:

- La ejecución de cada uno de los escenarios se realizó correctamente y el porcentaje de error en la ejecución de cada una de las transacciones fue mínimo. Se generaron las ilustraciones 77 y 78 como gráficas representativas del Escenario de Negocio:

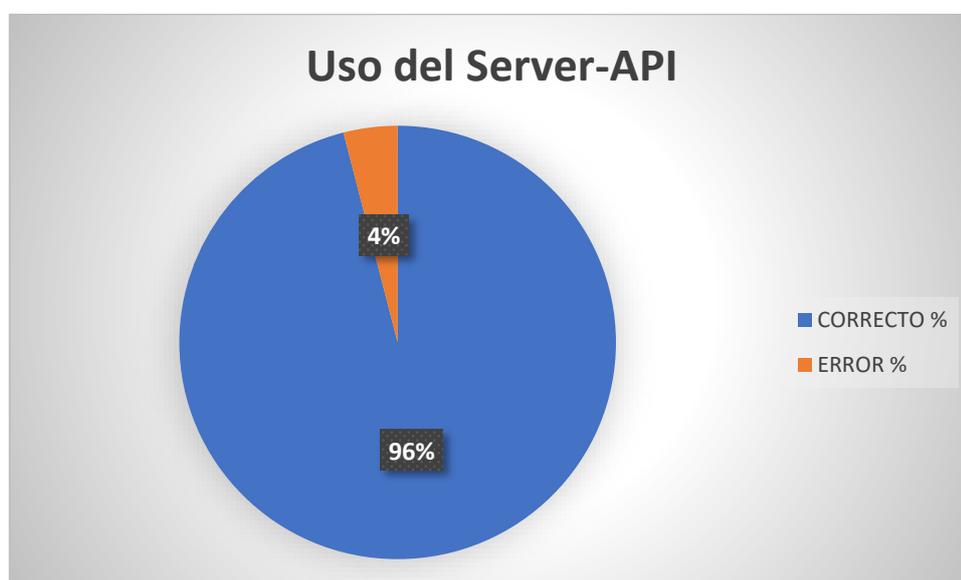


Ilustración 77. Porcentaje de solicitudes usando el Bot

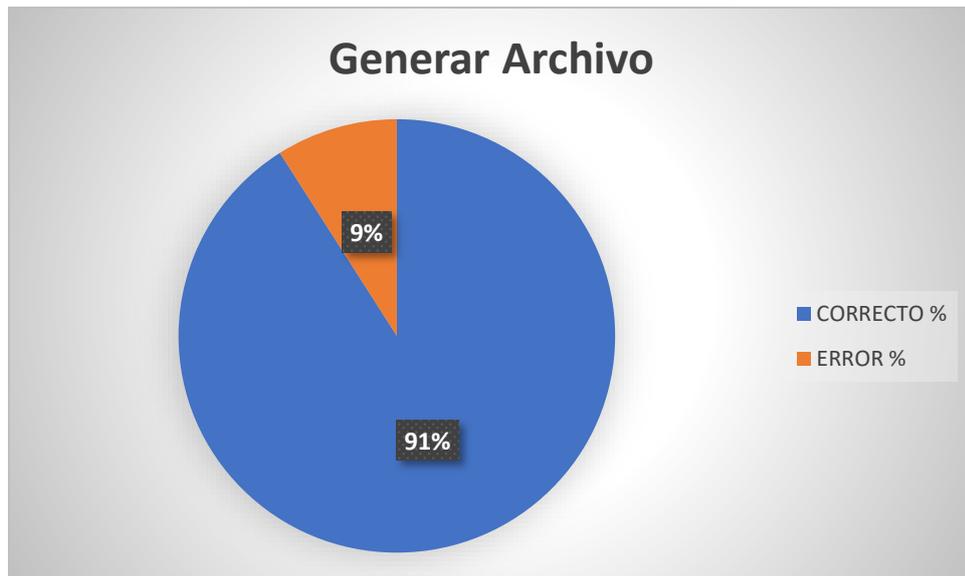


Ilustración 78. Porcentaje para la generación del formulario

- En la ilustración 79, se muestra el throughput de respuesta para el Escenario de negocio, en este escenario se simuló 120 usuarios concurrentes con el resultado de throughput de 3.5/seg para todas las peticiones realizadas. Se obtuvieron tiempos de respuesta aceptables, estos resultados dependen de la configuración y recursos de un entorno a otro.

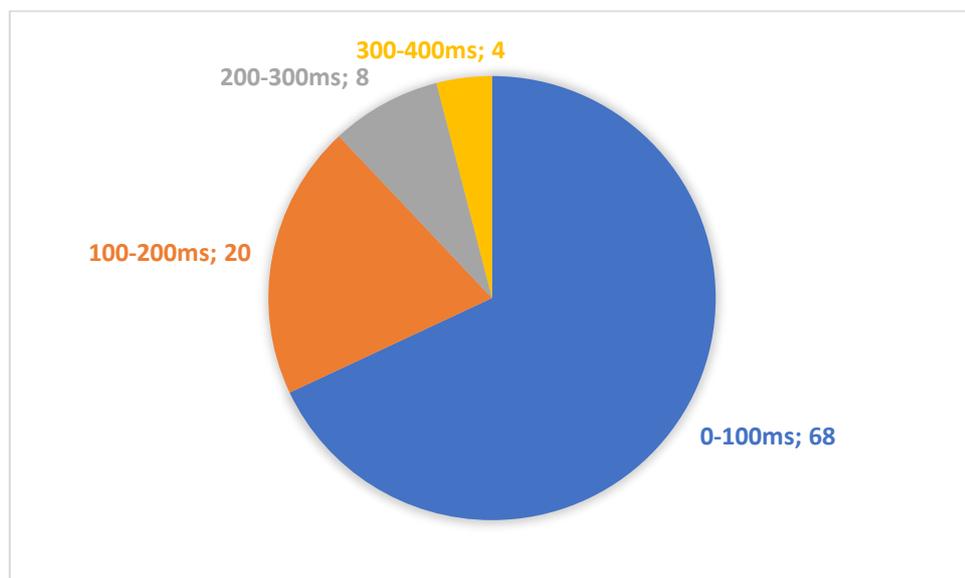


Ilustración 79. Tiempos de respuestas para las solicitudes

- El consumo de CPU y memoria permanecieron estables durante la ejecución de los escenarios de inicial y de negocio (4h y 8h respectivamente)

A continuación, se puede observar en la tabla 40, los indicadores que sirvieron como referencia para evaluar el grado de aceptación de la prueba:

Tabla 40. Indicadores de uso del hardware

Indicador	Valor
Tiempo medio de la transacción de generar formulario (120 usuarios concurrentes)	<0.5 segundos
Tiempo medio de respuesta a petición (120 usuarios concurrentes)	<0.5 segundos
Tiempo medio en generar documento (120 usuarios concurrentes)	<1 segundo
Tiempo medio de la transacción carga carrusel (120 usuarios concurrentes)	<0.5 segundos
Tiempo medio de la transacción carga respuesta con multimedia (120 usuarios concurrentes)	<0.5 segundos
Numero de documento en 5 minuto con 10 hilos en el motor JAVA	120

Tras la ejecución y análisis de los resultados obtenidos durante la ejecución de cada uno de los escenarios ejecutados en las pruebas de carga se visualizó un incremento en el uso del procesador.

Después de las pruebas se obtuvieron las siguientes conclusiones:

- El sistema se muestra estable para periodos elevados de tiempo, asegurándose estabilidad en el consumo de CPU, memoria y carga del servidor. Durante la ejecución de los escenarios de negocio e inicial no se observó ningún pico en los tiempos de respuesta, ni se registró ningún error grave, obteniéndose un 0% de errores en las transacciones ejecutadas.
- El sistema proporciono buenos resultados durante la prueba de carga, el resultado manejo un número de usuarios concurrentes similar a 120 efectivos y los tiempos medios de respuesta fueron inferiores a 1 segundo.
- En el escenario de saturación, se estableció como parámetro de saturación el aumento progresivo de solicitudes al sistema hasta llegar a los 15000. Este incremento repercutió en el aumento de los tiempos de ejecución, especialmente en la generación del formulario y el manejo de multimedios. Durante el tiempo de restauración del servicio se notó un aumento mayor a la mitad del uso de los procesadores, pero cuya duración es de apenas segundos, este proceso no afecta el uso cotidiano del servicio

Tras un estudio del comportamiento del SmartBot Nati, se concluyó que: “el sistema, muestra tiempos medios de respuestas muy similares a los esperados, obteniéndose un resultado satisfactorio en la ejecución de todas las pruebas”.

Para más información sobre las herramientas usadas y los parámetros evaluados (Ver anexo F)

4.4 Evaluación de índices de calidad y utilidad

Las aplicaciones fueron evaluadas a través de una encuesta (ver Anexo E) para obtener una estadística en función de la satisfacción y eficiencia. La encuesta fue aplicada a un grupo de usuarios pertenecientes a la institución después de usar los aplicativos.

4.4.1 Formulación de Encuesta

La encuesta estará dividida en 3 secciones, la primera para evaluar la efectividad del ChatBot y la segunda, para evaluar la satisfacción de usuario. La tercera evaluará edades de uso, sugerencias y cualidades atractivas al usuario

EFFECTIVIDAD *

	Deficiente	Medio	Satisfactorio	Muy bueno	Excelente
¿El SmartBot brinda ayuda con la necesidad solicitada por el usuario?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>
¿El tiempo de respuesta le pareció óptimo?	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>	<input type="radio"/>
¿La información presentada por el SmartBot fue correcta?	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input checked="" type="radio"/>	<input type="radio"/>

Ilustración 80. Preguntas usadas en la encuesta-sección efectividad

SATISFACCIÓN DEL USUARIO *

	Deficiente	Medio	Satisfactorio	Muy bueno	Excelente
En general, ¿Cómo te pareció el trato de Nati?	<input type="radio"/>				
¿Nati presento un dialogo cordial e intuitivo?	<input type="radio"/>				
¿Qué calificación le das a la interacción con Nati?	<input type="radio"/>				
¿Qué calificación le das a la captura de voz para Nati dentro de los kioskos?	<input type="radio"/>				
¿Crees que los elementos multimedia facilitaron la interacción con Nati?	<input type="radio"/>				
¿Crees que el sintetizador de voz mejora la interacción de uso?	<input type="radio"/>				

Ilustración 81. Preguntas usadas en la encuesta-sección satisfacción del usuario

4.4.2 Análisis de Resultados

La muestra se tomó desde el 20/10/2018 hasta el 12/11/2018 y corresponde a un total de 60 personas.

- Se observó que la mayor muestra se encuentra en edades comprendidas entre los 21 y 25 años.

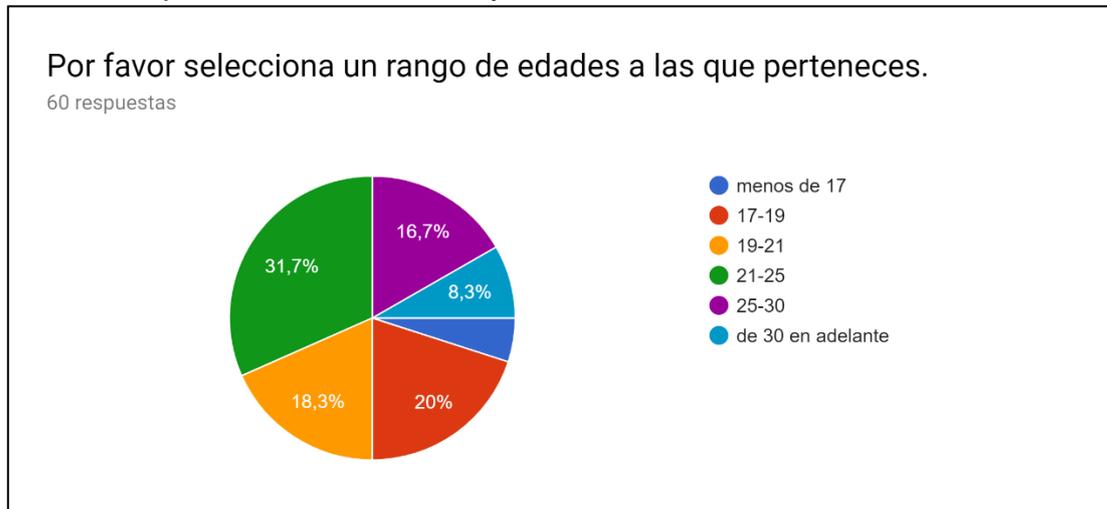


Ilustración 82. Gráfica pastel encuesta de edades

- Se observó que la mayor muestra se encuentra en edades comprendidas entre los 21 y 25 años.

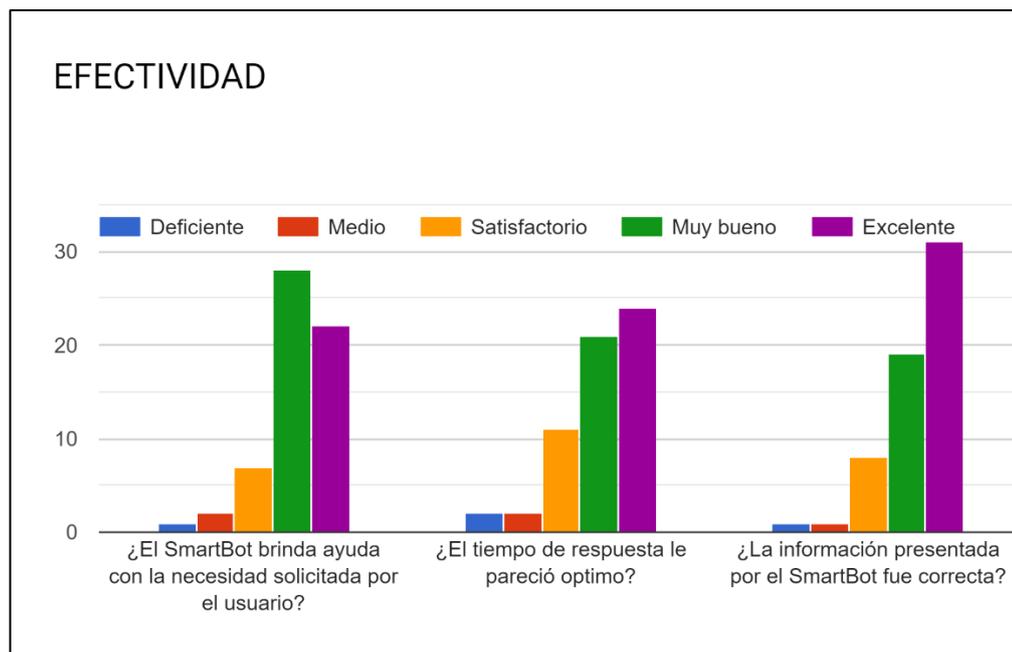


Ilustración 83. Gráficas de columna efectividad

- Se observó en la mayoría de los casos la aceptación del SmartBot en

promedio es muy buena, los resultados excelentes fueron: el uso en general y la captura de voz. Dentro de las gráficas se pudo apreciar que el módulo de sintetización de voz cuenta con aceptación del público, pero este módulo debe mejorar para adaptarse mejor a las necesidades del usuario, en este caso se debe tomar en cuenta los limitantes de tecnología con respecto a la emisión de voz robótica.

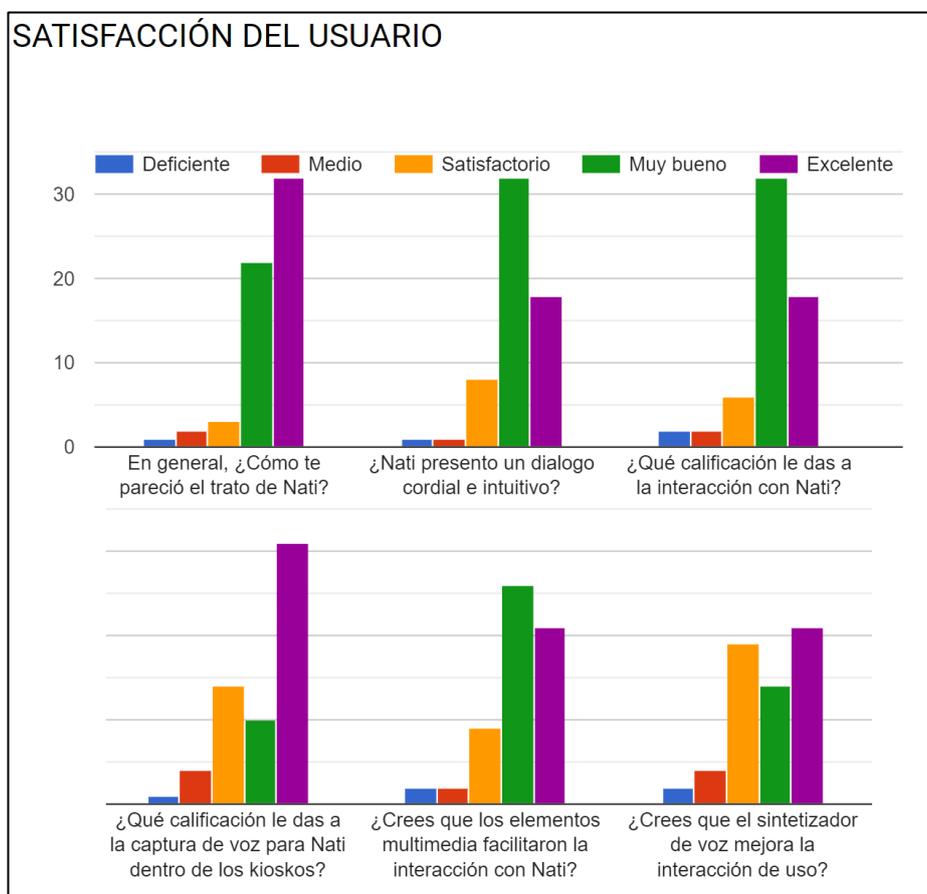


Ilustración 84. Graficas de columna satisfacción del usuario

- Dentro de las sugerencias para mejoras de Nati, el 75% de los usuarios sugirieron la complementación de información y aumento de temas que maneja Nati, mientras que un 10% sugirió la mejora a las respuestas entregadas.

Tabla 41. Valores estimados de la encuesta de satisfacción

¿Qué mejorarías de Nati?	N.º
Más información	75%
Sus respuestas	10%
La interfaz	8%
Interfaz	6%

En su mayoría los usuarios reportan que es necesario desarrollar nuevos contextos para el SmartBot, esto debido a que el proyecto tiene un alcance limitado, es importante indicar en la promoción de este que al inicio tendrá tópicos limitados y su alcance se irá ampliando en el tiempo, esto con la finalidad de no generar falsas expectativas. Del lado del desarrollo se está trabajando en mejorar el mensaje ante las preguntas no respondidas.

La mayoría de la información solicitada por los estudiantes se centró en bienestar estudiantil, directorios telefónicos y de direcciones, el resto de lo solicitado representó menor porcentaje en la muestra.

Es importante resaltar que la muestra estuvo limitada a estudiantes de carreras tecnológicas, por lo que se puede tener una muestra sesgada; para lo cual es recomendable incluir a personas de otras carreras y personal no perteneciente a la institución dentro de dichas pruebas.

Capítulo 5 Conclusiones y Recomendaciones

5.1 Conclusiones

Las empresas actualmente están innovando con los asistentes virtuales por ser una nueva iniciativa para brindar una mejor atención al usuario. Los asistentes han cambiado el uso de interfaces tradicionales de interacción entre el usuario y un computador, los asistentes se encargan de ejecutar una serie de tareas durante la conversación.

Para la implementación de un asistente virtual institucional es necesario entender lo que la institución quiere representar en el medio. Para esto es necesario el analizar los procesos dentro de la organización, comprender el trabajo y las expresiones usadas por el personal. Con este se buscó, cumplir con el objetivo general y se obtuvieron buenos resultados para la institución.

Gracias a los servicios cognitivos ofrecidos por DialogFlow se pudo simular una conversación más fluida y natural entre el asistente y el usuario, además se pudo establecer los limitantes y condiciones necesarias para el funcionamiento de IA con el uso de un perfil de personalidad.

La implementación de los asistentes virtuales dentro del proyecto, generan mejoras en los siguientes aspectos:

- ❖ Los usuarios de la página web tienen un nuevo medio para resolver sus inquietudes académicas y de infraestructura tecnológica.
- ❖ Los tiempos de respuesta a inquietudes se reducen y algunas inquietudes son contestadas de manera inmediata.
- ❖ Se genera más interés en la implementación de asistentes de otras áreas dentro de la institución.
- ❖ Las cargas de personal encargada de responder a las inquietudes de estudiantes y profesores disminuyen, dado que el usuario es guiado en el llenado de formularios y obtiene documentación sin la necesidad de comunicarse con un asesor real.

5.2 Recomendaciones

Una vez finalizada la etapa de prototipado, será necesario estimar las posibles mejoras de las funciones para poder permitir al asistente predecir respuestas a diálogos distintos a áreas asignadas inicialmente. Es recomendable estudiar e identificar un mecanismo que permita separar contextos no deseados del área de aprendizaje e igualmente de un mecanismo de sugerencia a una respuesta dada por DialogFlow.

El mejoramiento a las respuestas dadas por la herramienta DialogFlow depende del continuo entrenamiento y ajuste de la cantidad de mensajes ocupados para la preparación del dialogo, después de la implementación del asistente, se tomó en cuenta que durante el periodo de prueba y pase a

producción, existía un desbalance entre los mensajes por cada contexto esto puede conducir a confusión del algoritmo debido a que la carga de contextos tiende a ir hacia las áreas más usadas y los contextos poco usados pierden prioridad.

Después de la creación de este prototipo y su futuro desarrollo queda como propuesta para trabajos futuros la implementación de técnicas adicionales de procesamiento de lenguaje natural como son el uso de subsecuencias de n elementos o N-Gramas, el uso de bolsas de palabras, entre otras. Dentro del uso de clasificación de palabras se puede tomar en cuenta el uso de algoritmos como Naive Bayes el cual hace uso de teoría de la probabilidad y minería de datos.

Además, se podría adecuar el sistema de SmartBot, para poder definir tareas exclusivas por perfiles ej. Estudiantes, Profesores, Usuarios externos a la institución. Gracias a estos perfiles se podría definir mejor los contextos necesarios y priorizar necesidades durante el desarrollo y mejoramiento del actual y futuro asistente virtual.

Adicionalmente se debe tomar en cuenta que el desarrollo de un proyecto de construcción de un asistente virtual depende en gran medida de la identificación de los principales contextos de conversación, esto brindará los límites del proyecto, con esto se podrá elegir entre las distintas herramientas tecnológicas.

5.3 Posibles Trabajos Futuros

Integración con Sistema de administración estudiantil

El actual proyecto de titulación cuenta con contextos de ayuda de llenado de documentos y respuesta a solicitud de documentos con distintas temáticas, la integración con el sistema SAEw facilitaría la obtención de los datos personales y académicos del estudiante, esto podría reflejarse en mejoramiento de tiempos de llenado de los distintos documentos y también en mejoramiento de la interacción entre el sistema y sus distintos usuarios.

Adicionalmente el sistema SAEw podría contar con varios agentes conversacionales especializados para los distintos contextos e inclusive se podría tomar en cuenta las necesidades especiales de distintos usuarios como es el uso del habla y opciones animadas.

Animaciones 3D para el asistente virtual

El avatar o carácter animado brinda a la interacción entre la máquina y el usuario una interfaz más agradable e intuitiva, esto se ve reflejado en la satisfacción que tiene el usuario al percibir un estímulo antropomórfico brindado por una serie de animaciones 3D como se puede visualizar en la ilustración 85. Gracias a este estímulo y al ser considerado como un valor adicional innovador para los proyectos, el desarrollo de estos avatares ha avanzado a grandes pasos en los últimos años, los avatares han sido

integrados en el cine, televisión, juegos y muchas otras áreas relacionadas. [50]

Dentro del mercado del área tecnológica del habla se ha visto desplazada el desarrollo de la sinterización de voz para muchos idiomas, pero gracias a tecnologías de animación más accesible a los usuarios en los últimos tiempos, se ha notado un aumento en el catálogo disponible de lenguajes usados para el servicio de asistentes virtuales o interfaces conversacionales como lo son avatares dentro de un videojuego. [50]

Inclusive para futuras líneas de investigación se podría estudiar la sinergia existente entre las tecnologías de desarrollo gráfico y las tecnologías de desarrollo de sinterización de voz y sonidos, ya que juntas brindan al usuario un campo mucho más amplio en nuevos desarrollos como son la animación corporal y facial, el manejo de entonación conversacional y reacciones de los avatares a distintos entornos.

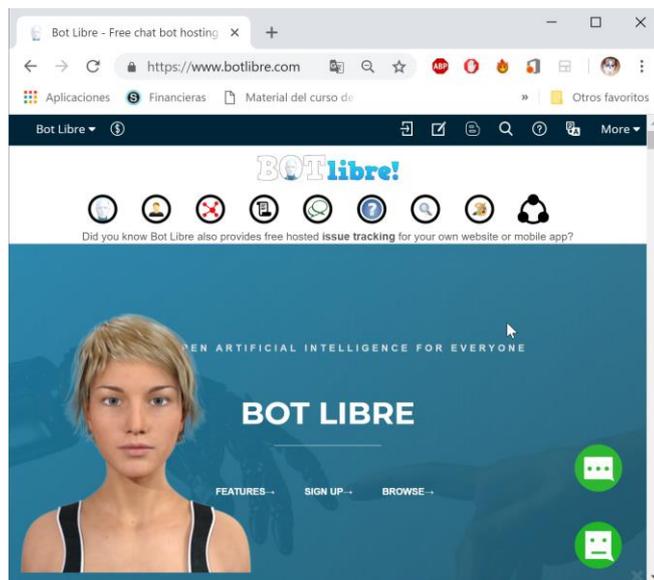


Ilustración 85. Uso de botlibre

Bibliografía

- [1] J. Medina, E. Eisman, J. Castro , *Asistentes virtuales en plataformas 3.0*, Granada - España, 2013.
- [2] J. Liou, *Toolbox for prototype development*, Taylor & Francis Group, 2008.
- [3] Maruti Techlabs, «Top 5 Benefits Of Using Chatbots For Your Business,» 7 noviembre 2017. [En línea]. Available: <https://chatbotsmagazine.com/top-5-benefits-with-using-chatbots-for-your-business-159a0cee7d8a>. [Último acceso: 1 noviembre 2018].
- [4] Forbes, «Thinking About Chatbots For Marketing,» 27 junio 2017. [En línea]. Available: <https://www.forbes.com/sites/forbescommunicationscouncil/2017/06/27/thinking-about-chatbots-for-marketing-consider-this/#2ff79d15718e>. [Último acceso: 21 enero 2018].
- [5] Dialogflow, «Uso del IA en Dialoflow» 4 enero 2017. [En línea]. Available: <https://dialogflow.com/>. [Último acceso: 2 noviembre 2018].
- [6] Woods, Abdul-Kader, «Survey on chatbot design techniques in speech conversation systems,» *International Journal of Advanced Computer Science and Applications*, vol. 6, nº 7, pp. 77-79, 2015.
- [7] J. Rodríguez, H. Merlino, E. Fernández, «Comportamiento Adaptable de Chatbots Dependiente del Contexto,» *Revista Latinoamericana de Ingeniería de Software*, vol. 2, nº 2, pp. 115-135, 2014.
- [8] A. Augello, G. Pilato, A. Machi, S. Gaglio , «An approach to enhance chatbot semantic power and maintainability: experiences within the FRASI project,» *Semantic Computing (ICSC)*, vol. IEEE Sixth International Conference , nº 2, pp. 186-193, 2012.
- [9] Torres, Cobos, «Integración de un Chatbot como habilidad de un robot social con gestor de diálogos,» de *Facultad de Ingeniería de Sistemas y Automática-Universidad Carlos III de Madrid*, Madrid, 2013.
- [10] P. Hingston, *Believable Bots - Can Computers play like people*, Mount Lawley: Springer, 2012.
- [11] S. Etlinger(Analyst), *The conversational business - How chatbots will reshape digital experiences*, California: Altimert, 2017.
- [12] Statista, «Number of mobile phone users,» 2018. [En línea]. Available: <https://www.statista.com/statistics/274774/forecast-of-mobile-phone-users-worldwide/>. [Último acceso: 04 noviembre 2018].
- [13] T. Welt , «De Audrey a Siri: historia y uso de los asistentes virtuales,» 31 08 2017. [En línea]. Available: <https://www.alisys.net/es/blog/95-de-audrey-a-siri-historia-y-uso-de-los-asistentes-de-virtuales> [Último acceso: 20 08 2018].
- [14] Pereira Juanan, Medina Haritz, Díaz Óscar., «Uso de Chatbots en la Docencia Universitaria,» 2016. [En línea]. Available: <http://romulo.det.uvigo.es/ticai/libros/2016/2016/Cap13.pdf>. [Último acceso: 02 09 2018].

- [15] P. Cornejo, «Arquitectura de un Chatbot,» 04 2106. [En línea]. Available: <https://medium.com/@patcornejo/arquitectura-de-un-chatbot-cb2d1c5f86c7>. [Último acceso: 15 09 2018].
- [16] J. Cassell, T. Bickmore , L. Campbell , H. Vilhjálmsón , H. Yan, «Conversation as a System Framework: Designing Embodied Conversational Agents,» 2000. [En línea]. Available: <http://citeseerx.ist.psu.edu/viewdoc/similar?doi=10.1.1.31.711&type=ab>. [Último acceso: 15 08 2018].
- [17] J. A. Ramírez, «El uso de las computadoras en el proceso de enseñanza-aprendizaje: un estudio de caso,» *Educación y ciencia*, vol. 7, nº 28, pp. 5-12, 2003.
- [18] A. Gelbukh, «Procesamiento de Lenguaje Natural y sus Aplicaciones,» *Komputer Sapiens*, vol. 2, nº 2, pp. 6-7, 2010.
- [19] A. C. Vásquez, J. P. Quispe, and A. M. Huayna, «Procesamiento de lenguaje natural,» *Revista de investigación de Sistemas e Informática*, vol. 6, nº 2, pp. 45-53, 2009.
- [20] A. Simon, M. Singh Deo, V. Selvam, and R. Babu, «An Overview of Machine Learning and its Applications,» pp. 22-24, 2016.
- [21] D. A. Zamora Ávila, «Métodos Machine Learning aplicados para estimar la concentración de los contaminantes de la DQO y de los SST en hidrosistemas de saneamiento urbano a partir de espectrometría UV-Visible,» 2013. [En línea]. Available: <http://hdl.handle.net/10554/3811>. [Último acceso: 01 09 2018].
- [22] Mastercard, «Securing Conversational Commerce: How to Make your Interactions with Virtual Assistants Safer,» 01 25 2018. [En línea]. Available: <https://newsroom.mastercard.com/2018/01/25/securing-conversational-commerce-how-to-make-your-interactions-with-virtual-assistants-safer/>. [Último acceso: 07 08 2018].
- [23] Futurizable, «Estado del arte en el desarrollo de chatbots a nivel mundial,» 22 septiembre 2017. [En línea]. Available: <https://futurizable.com/chatbot/>. [Último acceso: 04 noviembre 2018].
- [24] Microsoft - TechNet, «Microsoft Bot Framework Basics: Building Intelligent Bots - TechNet Articles,» 10 julio 2017. [En línea]. Available: <https://social.technet.microsoft.com/wiki/contents/articles/37731.microsoft-bot-framework-basics-building-intelligent-bots-part-1.aspx>. [Último acceso: 23 enero 2018].
- [25] IBM, «Watson AI Assistant,» 2018. [En línea]. Available: <https://www.ibm.com/watson/services/conversation/>. [Último acceso: 04 noviembre 2018].
- [26] Bot Libre, «SDK - Bot libre,» 26 julio 2018. [En línea]. Available: <https://www.botlibre.com/sdk.jsp?>. [Último acceso: 04 noviembre 2018].
- [27] Wit Inc, « Statistical Analysis for companies » 2018. [En línea]. Available: <https://wit.ai/>. [Último acceso: 04 noviembre 2018].
- [28] Microsoft Azure, «Use LUIS and QnA services with the Dispatch tool,» 30 octubre 2018. [En línea]. Available: <https://docs.microsoft.com/en-us/azure/bot-service/bot-builder-tutorial-dispatch?view=azure-bot-service-4.0&tabs=csharp>. [Último acceso: 09 noviembre 2018].

- [29] E. Dimitrova, «Bot Wars: Wit.ai vs Chatfuel,» 14 diciembre 2016. [En línea]. Available: <https://chatbotsmagazine.com/bot-wars-wit-ai-vs-chatfuel-5e978070b9e>. [Último acceso: 09 noviembre 2018].
- [30] A. Kulkarni, «Using Dialogflow with microsoft bot framework,» 06 diciembre 2016. [En línea]. Available: <https://chatbotsmagazine.com/using-api-ai-with-microsoft-bot-framework-577a5d9fb66f>. [Último acceso: 09 noviembre 2011].
- [31] Microsoft Azure, «Azure Bot Service pricing,» 2018. [En línea]. Available: <https://azure.microsoft.com/en-us/pricing/details/bot-service/>. [Último acceso: 09 noviembre 2018].
- [32] Microsoft Azure, «Azure Bot Service Documentation,» 2018. [En línea]. Available: <https://docs.microsoft.com/en-us/azure/bot-service/?view=azure-bot-service-4.0>. [Último acceso: 09 noviembre 2018].
- [33] N. Jhilam, «How to Identify the Right Platform for Your Chatbot,» 27 febrero 2018. [En línea]. Available: <https://chatbotsmagazine.com/identifying-the-platform-for-your-chatbot-312e46ef9e8a>. [Último acceso: 09 noviembre 2018].
- [34] IBM, «How to build a chatbot for your business,» 2018. [En línea]. Available: <https://www.ibm.com/watson/how-to-build-a-chatbot/>. [Último acceso: 09 noviembre 2018].
- [35] M. Stoica, M. Mircea y B. Ghilic-Micu, «Software Development: Agile vs. Traditional,» *Informatica Económica*, pp. 64-76, 2013.
- [36] K. Beck, «¿Qué es XP?,» 29 septiembre 1999. [En línea]. Available: http://software2012team23.googlecode.com/git-history/5127389d21813c2bd955c53999f66cede994578b/docs/literature/Extreme_Programming_Explained_Kent_Beck_1999.pdf.
- [37] Sutherland, K Schwaber , «The Definitive Guide to Scrum: The Rules of the Game,» 07 2013. [En línea]. Available: <http://www.scrumguides.org/docs/scrumguide/v1/Scrum-Guide-US.pdf>. [Último acceso: 01 07 2018].
- [38] W3Schools, «Node JS Introduction,» 2018. [En línea]. Available: https://www.w3schools.com/nodejs/nodejs_intro.asp. [Último acceso: 06 noviembre 2018].
- [39] freeCodeCamp, «Understanding How the Chrome V8 Engine Translates JavaScript into Machine Code,» 19 diciembre 2017. [En línea]. Available: <https://medium.freecodecamp.org/understanding-the-core-of-nodejs-the-powerful-chrome-v8-engine-79e7eb8af964>. [Último acceso: 06 noviem 2018].
- [40] O. Blancarte, «Introducción a NodeJS (JavaScript del lado del Servidor),» 19 mayo 2017. [En línea]. Available: <https://www.oscarblancarteblog.com/2017/05/29/introduccion-a-nodejs-2/>. [Último acceso: 06 noviembre 2018].
- [41] ORACLE CORPORATION, «Tecnología Java,» 10 2017. [En línea]. Available: https://www.java.com/es/download/faq/whatis_java.xml. [Último acceso: 20 10 2018].

- [42] W. BRAD y D. DAMSTRA, WORDPRESS. DISEÑO Y DESARROLLO, ANAYA MULTIMEDIA, 2013.
- [43] J. Acuña, «Características y tabla comparativa de los sistemas operativos móviles más usados,» 17 marzo 2017. [En línea]. Available: <https://jmacuna.tecnoblog.guru/2017/03/sistemas-operativos-moviles.html#comment-form>.
- [44] M. Báez, Á. Borrego, J. Cordero, L. Cruz, M. González y F. Hernández., «Introducción a Android,» *Madrid: Creative Commons la Universidad Complutense de Madrid*, 2007.
- [45] Sencha, «Sencha Touch,» 2018. [En línea]. Available: <https://www.sencha.com/products/touch/>. [Último acceso: 06 noviembre 2018].
- [46] D. Wells, «XP-User Stories,» 2009. [En línea]. Available: <http://www.extremeprogramming.org/rules/userstories.html>. [Último acceso: 12 09 2018].
- [47] M. Lacey, M. Cohn, J. Sutherland, J. Newkirk, G. Watts y B. Wilson, «Real Training, Coaching, Experience: Real AGILE,» *Mitch Lacey & Associates, Inc.*, 2015.
- [48] V. Ravi, «THE TROJAN RETROSPECTIVE-SmoothApps,» 23 12 2014. [En línea]. Available: smoothapps.com/index.php/2014/12/the-trojan-retrospective-from-cricket-to-conversations/. [Último acceso: 13 09 2018].
- [49] R. Chung, «Intro to DialogFlow(API.ai) / Wit.ai / Luis.ai,» 24 octubre 2017. [En línea]. Available: <https://www.slideshare.net/ryan/intro-to-dialogflowapi-ai-wit-ai-luis-ai>. [Último acceso: 23 septiembre 2018].
- [50] M. Carretero, D. Oyarzun, A. Ortiz, «Animación Facial y Corporal de Avatares 3D a partir de la edición e interpretación de lenguajes de marcas,» 01 2004. [En línea]. Available: https://www.researchgate.net/publication/228950279_Animacion_Facial_y_Corporal_de_Avatares_3D_a_partir_de_la_edicion_e_interpretacion_de_lenguajes_de_marcas. [Último acceso: 05 11 2018].
- [51] J. Arreola y M. Juan Carlos, «Las 7 tendencias tecnológicas del 2017,» 6 enero 2017. [En línea]. Available: <https://www.forbes.com.mx/las-7-tendencias-tecnologicas-del-2017/>. [Último acceso: 1 diciembre 2018].

Anexos

Anexo A

Documento manual del Usuario

Anexo B

Documento manual técnico de configuración servidor y plugin

Anexo C

Documento manual técnico de configuración Móvil

Anexo D

Tabla de categorización de contextos

Anexo E

Formulario de eficiencia y satisfacción del SmartBot

Anexo F

Informe de pruebas de estrés y rendimiento

Anexo G

Historias de usuario

Anexo H

Video de uso del app móvil