



Universidad Nacional Mayor de San Marcos

Universidad del Perú. Decana de América

Facultad de Ingeniería de Sistemas e Informática

Escuela Profesional de Ingeniería de Software

**Desarrollo de un bot para la mejora del control del
trabajo remoto de los empleados de una empresa
usando Slack**

TRABAJO DE SUFICIENCIA PROFESIONAL

Para optar el Título Profesional de Ingeniero de Software

AUTOR

Ernesto Danilo CARRANZA CHIARA

ASESOR

Robert Elías ESPINOZA DOMÍNGUEZ

Lima, Perú

2022



Reconocimiento - No Comercial - Compartir Igual - Sin restricciones adicionales

<https://creativecommons.org/licenses/by-nc-sa/4.0/>

Usted puede distribuir, remezclar, retocar, y crear a partir del documento original de modo no comercial, siempre y cuando se dé crédito al autor del documento y se licencien las nuevas creaciones bajo las mismas condiciones. No se permite aplicar términos legales o medidas tecnológicas que restrinjan legalmente a otros a hacer cualquier cosa que permita esta licencia.

Referencia bibliográfica

Carranza, E. (2022). *Desarrollo de un bot para la mejora del control del trabajo remoto de los empleados de una empresa usando Slack*. [Trabajo de suficiencia profesional, Universidad Nacional Mayor de San Marcos, Facultad de Ingeniería de Sistemas e Informática, Escuela Profesional de Ingeniería de Software]. Repositorio institucional Cybertesis UNMSM.

Metadatos complementarios

Datos de autor	
Nombres y apellidos	Ernesto Danilo Carranza Chiara
Tipo de documento de identidad	DNI
Número de documento de identidad	47988945
URL de ORCID	https://orcid.org/0000-0001-5438-6218
Datos de asesor	
Nombres y apellidos	Robert Elías Espinoza Dominguez
Tipo de documento de identidad	DNI
Número de documento de identidad	08136325
URL de ORCID	https://orcid.org/0000-0002-3456-8302
Datos del jurado	
Presidente del jurado	
Nombres y apellidos	Cesar Alberto Molina Neyra
Tipo de documento	DNI
Número de documento de identidad	40553679
Miembro del jurado 1	
Nombres y apellidos	Javier Elmer Cabrera Díaz
Tipo de documento	DNI
Número de documento de identidad	08692591
Miembro del jurado 2 (Asesor)	
Nombres y apellidos	Robert Elías Espinoza Dominguez
Tipo de documento	DNI
Número de documento de identidad	08136325
Datos de investigación	
Línea de investigación	No aplica

Grupo de investigación	No aplica
Agencia de financiamiento	Sin financiamiento
Ubicación geográfica de la investigación	País: Perú Departamento: Lima Provincia: Lima Distrito: Cercado de Lima Jr. Carlos Amezaga No. 375 Universidad Nacional Mayor de San Marcos Latitud: -12.0564232 Longitud: -77.0843327
Año o rango de años en que se realizó la investigación	2022
URL de disciplinas OCDE	2.02.04 -- Ingeniería de sistemas y comunicaciones https://purl.org/pe-repo/ocde/ford#2.02.04



UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
Escuela Profesional de Ingeniería de Software

Acta Virtual de Sustentación
del Trabajo de Suficiencia Profesional

Siendo las 21:00 horas del día 22 de agosto del año 2022, se reunieron virtualmente los docentes designados como Miembros de Jurado del Trabajo de Suficiencia Profesional, presidido por el Ing. Molina Neyra Cesar Alberto (Presidente), Mg. Cabrera Díaz Javier Elmer (Miembro) y el Ing. Espinoza Domínguez Robert Elías (Miembro Asesor), usando la plataforma Meet (<https://meet.google.com/qci-igcr-kdc>), para la sustentación virtual del Trabajo de Suficiencia Profesional intitulado: **“DESARROLLO DE UN BOT PARA LA MEJORA DEL CONTROL DEL TRABAJO REMOTO DE LOS EMPLEADOS DE UNA EMPRESA USANDO SLACK”**, por el Bachiller Carranza Chiara Ernesto Danilo; para obtener el Título Profesional de Ingeniero de Software.

Acto seguido de la exposición del Trabajo de Suficiencia Profesional, el Presidente invitó al Bachiller a dar las respuestas a las preguntas establecidas por los miembros del Jurado.

El Bachiller en el curso de sus intervenciones demostró pleno dominio del tema, al responder con acierto y fluidez a las observaciones y preguntas formuladas por los señores miembros del Jurado.

Finalmente habiéndose efectuado la calificación correspondiente por los miembros del Jurado, el Bachiller obtuvo la nota de **18 (DIECIOCHO)**

A continuación el Presidente de Jurados el Ing. Molina Neyra Cesar Alberto, declara al Bachiller **Ingeniero de Software**.

Siendo las **21:55** horas, se levantó la sesión.

Presidente

Ing. Molina Neyra Cesar Alberto

Miembro

Mg. Cabrera Díaz Javier Elmer

Miembro Asesor

Ing. Espinoza Domínguez Robert Elías



Universidad Nacional Mayor de San Marcos
Universidad del Perú. Decana de América
Facultad de Ingeniería de Sistemas e Informática
Escuela Profesional de Ingeniería Software

INFORME DE EVALUACIÓN DE ORIGINALIDAD
N° 018-EPISW-FISI-2022

1. Autoridad Académica que emite el Informe de Originalidad:	Directora de la Escuela Profesional de Ingeniería de Software
2. Apellidos y Nombres de la autoridad académica:	Dra. Nora Bertha La Serna Palomino
3. Operador del programa informático de similitudes:	Dra. Nora Bertha La Serna Palomino
3. Documento evaluado:	Tesis para Pregrado Título: “Desarrollo de un bot para la mejora del control del trabajo remoto de los empleados de una empresa usando Slack”
5. Autores del documento:	Carranza Chiara, Ernesto Danilo
6. Fecha de recepción de documento	Recepción: 10/09/2022
7. Fecha de aplicación del programa detector de similitudes:	Revisión: 10/09/2022
8. Software utilizado:	Turnitin
9. Configuración del programa detector de similitudes:	Excluye textos entrecuillados: Sí Excluye biografías: Sí Excluye cadenas menores a 40 palabras: Sí Otro criterio (especificar): No
10. Porcentaje de similitudes según programa detector de similitudes	Cuatro por ciento (4%)
11. Fuentes originales de las similitudes encontradas	Se adjuntan en 01 (una) foja al presente informe
12. Observaciones:	Ninguna
13. Calificación de originalidad i. Documento cumple criterios de originalidad, sin observaciones ii. Documento cumple criterios de originalidad, con observaciones iii. Documento no cumple criterios de originalidad.	Documento cumple criterio de originalidad, sin observación
14. Fecha del Informe:	14/11/2022

Dra. Nora Bertha La Serna Palomino
Directora (e) de la EPISW

Dedicatoria

Dedico este trabajo a mi familia. Ellos siempre me han apoyado a lo largo de mi vida y fueron los que más me motivaron para que me esfuerce por conseguir el título profesional.

Agradecimientos

Agradezco a mi asesor por su guía, consejos y tiempo brindado para la elaboración de este informe.

También quiero agradecer a la empresa Aynitech por permitirme usar la experiencia que tuve con ellos como base de este trabajo.

Y a mis profesores y compañeros de estudios que formaron parte de mi etapa formativa en la universidad.

UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS

FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

ESCUELA PROFESIONAL DE INGENIERÍA DE SOFTWARE

Desarrollo de un bot para la mejora del control del trabajo remoto de los empleados de una empresa usando Slack

Autor: Carranza Chiara, Ernesto Danilo

Asesor: Espinoza Dominguez, Robert Elías

Título: Trabajo de Suficiencia Profesional para optar el Título Profesional de Ingeniero de Software

Fecha: Mayo del 2022

RESUMEN

El presente trabajo de suficiencia profesional trata sobre el desarrollo de un Slack bot para uso interno de la empresa Aynitech. El Slack bot tuvo como meta principal cubrir la necesidad de tener una forma automatizada y eficiente de certificar que los empleados de la empresa cumplían su jornada laboral y sus actividades, ya que la empresa había adoptado la modalidad de trabajo remoto a raíz del aislamiento social por la COVID-19. Para el desarrollo del bot se empleó una metodología ideada por la empresa junto con el framework Bolt de Slack, y otras tecnologías y herramientas open source. Además, el área de soporte de TI acondiciono un servidor en la empresa para hacer el despliegue de la Slack app como un software on-premise. La implementación de este bot tuvo como resultado más resaltante una reducción drástica del tiempo requerido para obtener la información de asistencia, lo que ayudo al área de recursos humanos a mejorar su productividad.

Palabras claves: Slack, bot, trabajo remoto, recursos humanos

MAJOR NATIONAL UNIVERSITY OF SAN MARCOS

FACULTY OF SYSTEMS AND INFORMATIC ENGINEERING

PROFESSIONAL SCHOOL OF SOFTWARE ENGINEERING

Development of a bot for the improvement of remote work monitoring of employees in a company using Slack

Author: Carranza Chiara, Ernesto Danilo

Advisor: Espinoza Dominguez, Robert Elías

Title: Professional Sufficiency Work to opt for the Professional Title of Software Engineer

Date: May 2022

ABSTRACT

This work of professional sufficiency is about the development of a Slack bot for internal use of the company Aynitech. The main goal of the Slack bot was to cover the need of having an automated and efficient way of certifying that the company's employees fulfilled their working hours and activities, since the company had adopted the remote work modality due to the social isolation by the COVID-19. For the development of the bot, a methodology devised by the company along with Slack's Bolt framework and other open source technologies and tools were used. In addition, the IT support area set up a server in the company to deploy the Slack app as an on-premise software. The implementation of this bot had as the most outstanding result a drastic reduction of the time needed to obtain attendance information, which helped the human resources area to improve its productivity.

Key words: Slack, bot, remote work, human resources

INDICE GENERAL

INTRODUCCIÓN	1
CAPÍTULO I TRAYECTORIA PROFESIONAL.....	3
1.1. Presentación	3
1.2. Experiencia laboral.....	3
1.3. Formación Académica.....	4
CAPÍTULO II CONTEXTO EN EL QUE SE DESARROLLA LA EXPERIENCIA	5
2.1. Empresa – Actividad que realiza	5
2.2. Visión	5
2.3. Misión.....	6
2.4. Organización de la empresa	6
2.5. Área, cargo y funciones desempeñadas	6
2.6. Experiencia profesional realizada en la organización	7
CAPÍTULO III ACTIVIDADES DESARROLLADAS	8
3.1. Situación Problemática	8
3.1.1. Definición del problema.....	9
3.2. Solución	10
3.2.1. Objetivos.....	10
3.2.2. Alcance.....	10
3.2.3. Etapas y Metodología.....	11
3.2.4. Fundamentos Utilizados.....	12
3.2.5. Implementación de las áreas, procesos, sistemas y buenas practicas	16
3.3. Evaluación	33
3.3.1. Evaluación Económica/ Evaluación Costo-Beneficio	33
3.3.2. Interpretación Del VAN y Del TIR.....	34
CAPÍTULO IV REFLEXIÓN CRÍTICA DE LA EXPERIENCIA	36
CAPÍTULO V CONCLUSIONES Y RECOMENDACIONES	37
5.1. Conclusiones	37
5.2. Recomendaciones.....	38
5.3. Fuentes De Información.....	39

5.4. Glosario	40
ANEXOS	41
Anexo 1: Cronograma de actividades del proyecto	41
Anexo 2: Cronograma de tareas del bot de reporte de tareas y de la aplicación web de gestión de tareas	42
Anexo 3: Prototipos construidos utilizando Block Kit Builder	43
Anexo 4: Manual de usuario del bot	44

INDICE DE FIGURAS

Figura 1: Organigrama de la empresa.....	6
Figura 2: Registro de asistencia antes de implementación del bot	9
Figura 3: Etapas del desarrollo para la primera parte de la aplicación.....	12
Figura 4: Estructura del código para la primera parte del bot	17
Figura 5: Herramienta de Slack para la construcción de prototipos: Block Kit Builder	18
Figura 6: Prototipo de formulario para la marcación de asistencia	18
Figura 7: Prototipo de notificación de asistencia.....	19
Figura 8: Prototipo del Home con funciones para RRHH.....	19
Figura 9: Prototipo del formulario de consulta de asistencia por fechas	20
Figura 10: Prototipo de notificación de consulta de asistencia	20
Figura 11: Diagrama de datos.....	21
Figura 12: Creación del Slash Command: /wfh.....	23
Figura 13: Menú para la asignación de <u>Scopes</u>	24
Figura 14: Configuración de Interactivity en el bot	24
Figura 15: Diagrama de flujo del funcionamiento del comando wfh.....	26
Figura 16: Diagrama de flujo del funcionamiento de la consulta de asistencia.	28
Figura 17: Diagrama de flujo del funcionamiento del comando /task	31
Figura 18: Diagrama de arquitectura del bot y la aplicación web	32
Figura 19: Cronograma del desarrollo de la primera versión.....	41
Figura 20: Cronograma de tareas de la segunda parte del proyecto	42
Figura 21: Payload en formato JSON de la interfaz de marcación de asistencia	43
Figura 22: Manual de usuario del bot.....	44

INDICE DE TABLAS

Tabla 1: Experiencia laboral.....	3
Tabla 2: Formación académica.....	4
Tabla 3: Lista de requisitos de la primera parte del proyecto.....	16
Tabla 4: Tecnologías y herramientas utilizadas en el desarrollo del bot.....	22
Tabla 5: Tecnologías y herramientas utilizadas en la construcción del componente web ...	30
Tabla 6: Costo del desarrollo del bot de registro de asistencia	33
Tabla 7: Costo del desarrollo del bot y de la aplicación web para la segunda parte	33
Tabla 8: Costo del registro de asistencia antes de la implementación del bot.....	34
Tabla 9: Flujo de caja	34
Tabla 10: VAN y TIR al sexto mes	34

INTRODUCCIÓN

Este trabajo de suficiencia profesional trata acerca de la implementación de un Slack bot para dar solución a los problemas detectados en el control del trabajo remoto que se originaron cuando la empresa adopto una política de trabajo remoto en respuesta a la emergencia sanitaria por la COVID-19. El principal problema que se identifico fue que la empresa carecía de una forma eficiente de validar que los horarios de la jornada diaria de cada empleado se estaban cumpliendo, ya que el anterior sistema que se utilizaba para este fin cayo en desuso debido a que solo podía funcionar de manera presencial pues dependía de un lector de huellas digital. La falta de este sistema, ocasiono que el área de recursos humanos realice el cálculo de la asistencia de forma manual, lo cual tomaba un aproximado de 3 horas diarias a la semana. Es importante resaltar que la información de asistencia de los empleados es de suma importancia pues permite calcular las remuneraciones, los descuentos, las vacaciones y esto no solo es crítico para el área de recursos humanos, sino que también afecta otras áreas como administración y contabilidad.

Esta situación motivo a la empresa a buscar una solución, la cual fue planteada por el área de soporte de TI con la cual el autor de este trabajo colaboró en tareas de definición de actividades, estimación de duración y esfuerzo de actividades y desarrollo.

Para la solución se planteó como objetivo hacer que la aplicación desarrollada haga que el proceso de registro de asistencia y actividades de los empleados de la empresa sea eficiente. Para ello se tomaron en cuenta que herramientas estaban disponibles en la empresa y podían aprovecharse. Una de ellas era Slack, una aplicación de mensajería para organizaciones, que la empresa ya había adoptado años atrás y utilizaba como canal de comunicación en la organización. Slack ofrece la posibilidad de crear aplicaciones personalizadas que se acomodan a las necesidades de las organizaciones y también tiene sus propias herramientas y frameworks para la construcción de estas aplicaciones.

Uno de los requisitos para la creación del bot es el de proveer un entorno donde alojar el código, para lo cual es ideal la nube, pero por un tema de costos y a que también se contaba con un servidor en las instalaciones de la empresa, se optó por que el Slack bot corra como una aplicación on-premise en dicho servidor.

Este documento se encuentra organizado de la siguiente manera:

En el CAPÍTULO I, se expone la trayectoria profesional dando detalle de cada una de las experiencias que hemos tenido durante nuestra carrera.

En el CAPÍTULO II, explicamos el contexto en el cual desarrollamos el presente trabajo profesional, esto incluye la mención de aspectos sobre la empresa donde se realizó la experiencia y la descripción los cargos ocupados y funciones realizadas.

En el CAPÍTULO III, hablamos de la situación del problema y su definición. Aquí también se detallan la solución y sus objetivos incluyendo las metodologías y herramientas utilizadas.

En el CAPÍTULO IV, se describe la reflexión crítica de la experiencia que tuvimos y damos detalles de lo aprendido durante el trabajo, como también se menciona las limitaciones que surgieron en el trabajo.

En el CAPÍTULO V, exponemos las conclusiones en base a los objetivos planteados anteriormente y presentamos algunas recomendaciones para este trabajo.

CAPÍTULO I TRAYECTORIA PROFESIONAL

1.1. Presentación

Bachiller en Ingeniería de Software de la Universidad Nacional Mayor de San Marcos con 5 años de experiencia en el diseño, desarrollo y mantenimiento de aplicaciones del lado del servidor haciendo uso principalmente de lenguajes de scripting y metodologías ágiles. A lo largo de su carrera ha construido soluciones tecnológicas para empresas pertenecientes a sectores como E-commerce, Medio Ambiente, Retail y TI.

1.2. Experiencia laboral

A continuación, se muestra la experiencia laboral del autor en la tabla 1.

Tabla 1
Experiencia laboral

Cargo y Empresa	Actividades realizadas	Periodo
Developer Aynitech S.A.C.	Cliente: Cliente en USA <ul style="list-style-type: none">▪ Desarrollo de un script que automatiza el archivado de canales inactivos de Slack utilizando Node.js, Lambda y S3.▪ Implementación de un Slack bot que se integra con ServiceNow usando Node.js, Serverless, Lambda y API Gateway.▪ Creación de un chatbot en Slack que ayuda con problemas relacionados a Help Desk. Uso de Python, Lambda y Amazon Lex.▪ Construcción de una integración entre Slack y Jira. Se utilizó Node.js, Nginx, Docker y Docker Compose.▪ Desarrollo de un Slack bot que automatiza el proceso de aprobación para la instalación de Slack apps. Construido con Typescript, Lambda, S3 y API Gateway.	Dic 2018 – Jun 2022

	Cliente: Aynitech	
	<ul style="list-style-type: none"> Construcción de un Slack bot para el registro de asistencia y actividades de los empleados de la empresa. 	
Python Developer Enova Agency S.A.C.	<ul style="list-style-type: none"> Construcción de 3 sitios web: jockey.com.pe, runastore.pe y zonacel.com usando Django, PostgreSQL y AWS. Desarrollo de integraciones con APIs de terceros para realización de pagos y servicios de entrega de pedidos. 	May 2018 – Nov 2018
Python Developer KashIO INC	<ul style="list-style-type: none"> Diseño y desarrollo de una REST API usando Flask, MySQL, Swagger y un enfoque API First. Construcción de un robot con Automate, una herramienta para RPA. Desarrollo de una integración con una API para la gestión de imágenes. 	Jun 2017 – Abr 2018
Fullstack developer Ecocentury S.A.C.	<ul style="list-style-type: none"> Diseño y desarrollo de una aplicación web para mejorar los procesos de registro de asistencia y control de logística usando el stack MEAN. 	Mar 2015 – Dic 2015
Web developer Intex Maquinarias S.A.C.	<ul style="list-style-type: none"> Desarrollo de nueva funcionalidad y mantenimiento de un e-commerce construido con Django, AngularJS, MySQL y AWS. Integración con la API de Google Maps para usar geolocalización. 	May 2014 – Feb 2015

Nota. Fuente: Elaboración propia

1.3. Formación Académica

En la tabla 2 se presenta la formación académica del autor.

Tabla 2
Formación académica

Periodo	Grado Académico	Institución Educativa
2009 - 2014	Bachiller en Ingeniería de Software	Universidad Nacional Mayor de San Marcos

Nota. Fuente: Elaboración propia

CAPÍTULO II CONTEXTO EN EL QUE SE DESARROLLA LA EXPERIENCIA

2.1. Empresa – Actividad que realiza

La empresa Aynitech realiza actividades de tecnología de la información y de servicios informáticos.

Según la página web de la empresa, esta es peruana y tiene 10 años de experiencia en consultoría de Business Analytics y Tecnologías de la información. Recientemente, la empresa se convirtió en un holding o sociedad gestora, dividiendo sus unidades de negocios en 5 compañías: una compañía que ofrece un EPM basado en la nube que integra inteligencia y analítica de negocios con sistemas transaccionales para permitir una transformación digital con cero inversión y bajos costos recurrentes, una compañía que tiene como principal producto una herramienta para análisis de redes sociales, una compañía que incluye que presta servicios BPO de ingeniería y de fábrica de Software, una compañía ofrece servicios de ciberseguridad a pequeñas y medianas empresas y otra compañía que provee servicios de Data Science, Analytics, Big Data y Machine Learning.

La empresa actualmente cuenta con más de 200 profesionales ubicados en sus oficinas de Austin, Santiago y Lima.

2.2. Visión

Convertirse en el mejor holding de Analítica, SaaS y BPO en América y hacer que la tecnología sea accesible a pequeñas y medianas empresas a un bajo costo, permitiendo a nuestros clientes una transformación digital eficiente para volverse exitosos en este nuevo mundo.

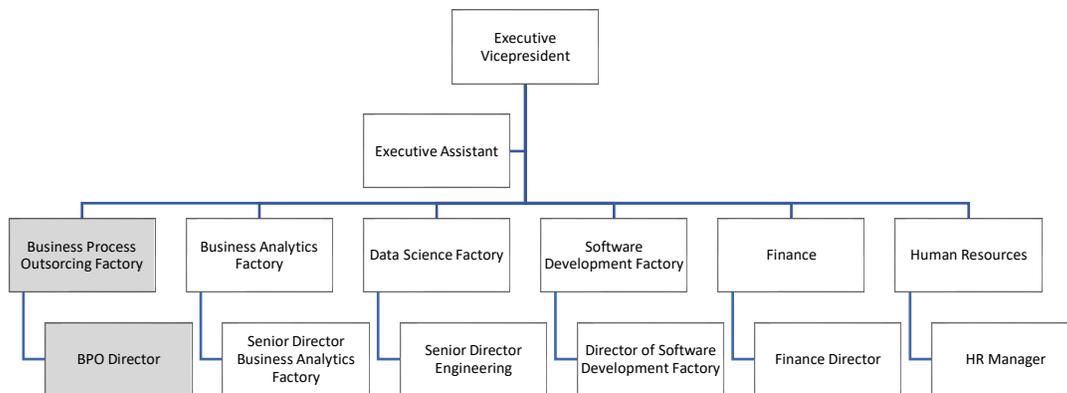
2.3. Misión

Hacer a nuestra gente los empleados más felices de Latino América, a la vez que ofrecemos a nuestros clientes los mejores servicios de Analítica e Ingeniería.

2.4. Organización de la empresa

En el organigrama de la figura 1 se muestra la estructura organizacional de la empresa, donde esta resaltada el área de BPO, a la cual el autor del trabajo perteneció.

Figura 1
Organigrama de la empresa



Nota. Fuente: Adaptación del organigrama de la empresa.

2.5. Área, cargo y funciones desempeñadas

2.5.1. Área

El autor del presente trabajo integró el área de BPO de una de las unidades de negocio de la empresa.

2.5.2. Cargo

El autor de este trabajo se desempeñó en el cargo de Developer en el área de BPO de una de las unidades de negocio de la empresa.

2.5.3. Funciones desempeñadas

Las funciones del cargo de Developer incluyen:

- Desarrollo de herramientas y aplicaciones según los requerimientos del cliente.
- Mantenimiento y corrección de bugs a las aplicaciones desarrolladas y existentes.
- Elaboración de documentación de las herramientas y aplicaciones construidas.
- Participación en estimación de esfuerzo y definición de tareas a realizar.
- Colaboración con otros desarrolladores para llevar a cabo revisiones de código.

2.6. Experiencia profesional realizada en la organización

A lo largo de nuestra permanencia en la empresa tuvimos el rol de Developer, lo que significó participar en el desarrollo de herramientas y aplicaciones usando la plataforma Slack, que el cliente ubicado en USA utilizaba. Así mismo, también se participó en el desarrollo de soluciones basadas en Slack para la empresa local en Perú. Es en estos desarrollos para la empresa local que surgió la oportunidad de llevar a cabo la experiencia profesional sobre la cual trata este trabajo. Para ello fue de mucha utilidad lo aprendido al trabajar para el cliente en USA, pues se aplicó la misma tecnología que ellos usaban para el desarrollo de aplicaciones de Slack.

CAPÍTULO III ACTIVIDADES DESARROLLADAS

3.1. Situación Problemática

Hasta inicios del año 2020, la empresa realizaba el proceso de registro de asistencia de los empleados utilizando un lector biométrico de huellas digitales ubicado en la entrada de cada una de sus oficinas. Adicionalmente, la empresa usaba como canal de comunicación corporativo la herramienta Slack. Durante el estado de emergencia por la COVID-19, se decretó el aislamiento social obligatorio que se puso en rigor a partir del 16 de marzo del 2020, dicho esto, el acceso a las oficinas de la empresa se vio impedido y por ello el sistema de registro de asistencia en funcionamiento no podía utilizarse, lo que devino en un cambio del proceso de registro de asistencias que en adelante se realizaría de manera manual, a través de la revisión y conteo de mensajes enviados por Slack o correos que los empleados usaban para hacer constar su asistencia.

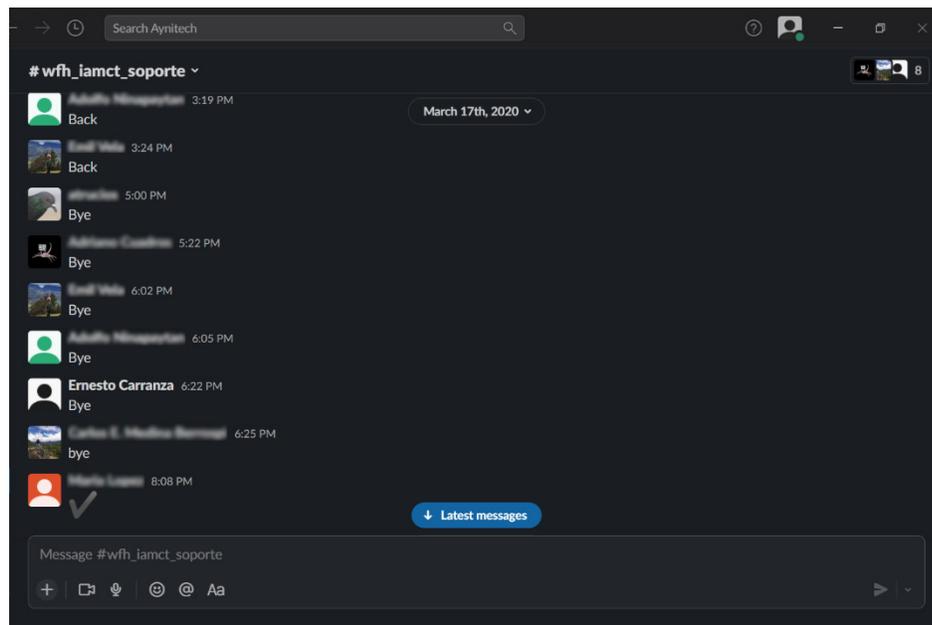
Los encargados en obtener la información de asistencia en la empresa era el personal del área de Recursos Humanos, que para llevar a cabo dicha tarea al final del día debían revisar los mensajes dejados en Slack y anotar las horas de inicio de la jornada, inicio del refrigerio, fin del refrigerio y fin de la jornada en una hoja Excel para cada uno de los 200 empleados, lo que podía tomar 3 horas diarias de trabajo.

Además, la información de asistencia de los empleados que se encuentra en Slack tiene un tiempo limitado para consultarse debido a que la empresa solo cuenta con el plan gratuito de Slack que permite acceso únicamente a los 10,000 mensajes más recientes.

También algunos clientes solicitaban a la empresa información referente a las actividades de los empleados que trabajaban para ellos, pero debido a que no se tenía esta información por no contar con ninguna herramienta para el seguimiento de tareas solo se podía otorgar estimados de esas actividades.

En la figura 2 se muestra como los empleados hacían constar las horas de su jornada laboral mediante el envío de mensajes a un canal de Slack antes de la implementación del bot.

Figura 2
Registro de asistencias antes de la implementación del bot



Nota. Fuente: Imagen tomada del Slack de Aynitech

3.1.1. Definición del problema

El registro de las asistencias de los trabajadores de la empresa es un proceso que puede tomar 3 horas diarias al área de recursos humanos debido a que no se encuentra correctamente automatizado. Además, hay una deficiencia en el seguimiento a las tareas realizadas por los empleados de Aynitech porque tampoco se cuenta con una manera automática de realizar este proceso.

3.2. Solución

Se desarrolló un Slack bot que permite a los empleados el registro de sus horas de inicio y fin de actividades, así como también las horas de refrigerio. Además, recursos humanos puede consultar esta información a través del bot. El bot también tiene una opción para que los colaboradores puedan registrar las tareas que hicieron durante el día. Estas tareas pueden gestionarse en una aplicación web que se integra con el bot.

3.2.1. Objetivos

Objetivo General.

Desarrollar un bot que permita mejorar el control del trabajo remoto de los empleados de la empresa Aynitech utilizando la plataforma Slack.

Objetivos Específicos.

- Desarrollar un bot con funcionalidad para que los empleados guarden las horas de su jornada laboral, además el bot debe proveer de un método al área de RRHH para la consulta de la información del fichaje de horas de los empleados.
- Desarrollar nuevas capacidades en el bot existente para que los empleados puedan reportar sus actividades diariamente a través de Slack. El Slack bot debe integrarse con una aplicación web donde las actividades reportadas se podrán gestionar.

3.2.2. Alcance

Alcance Funcional.

Se desarrolló un bot para la mejora del control del trabajo remoto en la empresa.

En el alcance funcional se incluye:

- Un bot construido en la plataforma Slack.
- Creación de un proceso para el reporte de tareas.
- No se incluyen las adaptaciones necesarias para integrar el bot con el sistema que procesa la información de asistencias, ya que este es mantenido por el área de Business Analytics.
- No se incluye el desarrollo de la aplicación web que servirá para gestionar las tareas reportadas, pero si es parte del alcance la integración del bot con esta aplicación web.

Alcance Organizacional.

El alcance abarca todas las áreas de la empresa, especialmente a Recursos Humanos, Business Analytics y BPO. El bot desarrollado impacta a todos los empleados de la empresa.

Alcance Geográfico.

El uso del bot se dará desde múltiples locaciones pues los colaboradores utilizarán el bot desde donde se encuentren y lo harán a través de Slack, aplicación a la cual se puede acceder mediante cualquier dispositivo con acceso a internet.

3.2.3. Etapas y Metodología

Etapas.

El desarrollo de la aplicación se dividió en dos partes: la primera parte consto en la creación del bot con la funcionalidad para el registro y visualización de las asistencias y en la segunda parte se agregó al bot la opción para el registro de tareas.

En la primera parte se puede distinguir las siguientes etapas:

- **Recopilación y definición de requisitos:** En esta etapa se tuvo reuniones con el equipo del proyecto, las áreas de RRHH y de Business Analytics. Los requisitos recopilados se evaluaron y los que fueron considerados se definieron.
- **Diseño de la estructura del código, de las interfaces y de los modelos del bot:** Para esta etapa se elaboró los prototipos de la interfaz del bot, también se escogió una estructura de código para la construcción del bot y se modelo la información necesaria.
- **Desarrollo del bot:** Esta es la etapa de programación, en la que utilizando las tecnologías seleccionadas se codifico el bot.
- **Pruebas al bot:** En esta etapa se realizaron pruebas de sistema al bot en un entorno de desarrollo.
- **Mantenimiento del bot:** Esta etapa incluye todos los cambios necesarios en el bot luego del lanzamiento inicial. Se realizaron cambios para corregir errores, agregar mejoras al código e implementar nueva funcionalidad.

A modo de resumen, se exponen estas etapas en la figura 3:

Figura 3
Etapas del desarrollo para la primera parte de la aplicación



Nota. Fuente: Elaboración propia

Metodología.

La metodología que se utilizó es propia de la empresa y se apoya en el uso de una herramienta de cronograma de actividades, en la cual se listan las tareas a realizar junto con su fecha de inicio y de fin.

3.2.4. Fundamentos Utilizados

Slack.

Slack es una aplicación de mensajería empresarial que permite a los miembros de una organización conectarse al conocimiento de la empresa. Slack cambia el modo en que colaboran los miembros de las organizaciones al reunirlos en un único espacio de trabajo. (SLACK, 2022)

Según lo mencionado por Markovich (2021), Slack es una plataforma de comunicación para negocios que permite a personas y empresas comunicarse a través de canales e hilos para obtener un entorno de trabajo simple y eficiente.

Ventajas y desventajas de Slack

Para Markovich (2021), entender las ventajas y desventajas de Slack nos permitirá saber de manera más sencilla si esta herramienta será una buena opción para la comunicación en nuestra organización.

Ventajas

- **Interfaz de usuario:** Una de las principales ventajas de Slack es que su interfaz cuenta con un diseño amigable para los usuarios.
- **Conexiones rápidas:** Slack cuenta con funciones para el envío de mensajes grupales e individuales. El envío de mensajes es rápido y eficiente, lo que ayuda a tener un alto grado de comunicación en toda la empresa.
- **Búsqueda:** Slack guarda todas las conversaciones que se dan en la plataforma, y permite encontrarlas de manera inmediata.
- **Integraciones:** Otra ventaja importante de Slack, es su capacidad para integrarse con otras herramientas, aplicaciones y bots.

Desventajas

- **Incomodidad:** Slack puede sentirse como una aplicación de chat, lo que causa agrado a algunas personas y molestia en otras. La facilidad para enviar mensajes a través de la plataforma puede ocasionar comunicaciones fuera del horario de oficina. Además, las notificaciones que Slack envía pueden terminar hostigando a los usuarios.
- **Desorganización:** La facilidad y rapidez para comunicarse de Slack puede hacer complicado a los usuarios mantenerse al tanto de las conversaciones. Esto es peor si solo se cuenta con el plan gratuito.

Comparación con otras herramientas

Microsoft Teams

Teams de Microsoft es una plataforma empresarial para comunicación y colaboración y es el principal competidor de Slack. De acuerdo a Markovich (2021), la principal ventaja de Teams sobre Slack es su capacidad de integración con los servicios de Microsoft, lo que lo hace una opción atractiva para equipos que cuentan con suscripción a Office 365.

Teams fue ideado para empresas grandes, es por eso que no es muy amigable para negocios pequeños, start-ups en etapa inicial y emprendedores. A diferencia de Slack, que se adapta a las necesidades de empresas de cualquier tamaño.

Otras herramientas

En el mercado de Slack también podemos encontrar los siguientes competidores:

- **Workplace by Facebook:** Una herramienta para la colaboración de equipos con un parecido a una red social.
- **Google Workspace:** Es una suite de productos Google como Gmail, Meet, Calendar, Drive, Docs, Sheet y otros que agregan características para empresas como dirección de correo personalizado, almacenamiento ilimitado y soporte por teléfono y correo electrónico.

Slack apps

Slack apps son un tipo de aplicación que pueden conectar una herramienta o servicio como Jira, Trello o Drive con Slack. Las apps que se integran a Slack permiten que las funcionalidades de diversos servicios puedan utilizarse directamente desde Slack.

Las Slack apps tienen acceso a la API de Slack, lo que les permite realizar operaciones de lectura, escritura, y actualización sobre todo el contenido de la plataforma. (SLACK, 2022).

Slack API

Slack proporciona diferentes tipos de APIs a las que puede accederse para interactuar con la plataforma. Cada API tiene una finalidad concreta, así como ventajas y desventajas. Los siguientes componentes son parte de la API de Slack: (Monjardín et al, 2018)

- **Web API:** Esta API provee una colección de métodos HTTP que constituyen la mayoría de funcionalidad que posee una Slack app.
- **Events API:** La API de Events permite a las apps responder a actividades en Slack.
- **RTM API:** Una API ya obsoleta que otorga una conexión WebSocket para acceder algunas de las mismas funciones que dan las APIs Web y de Events.

Bots.

Shevat (2017) señala que los bots son una nueva interfaz de usuario, que permite la interacción con funcionalidad software a través de aplicaciones de mensajería. Los bots también son conocidos como chatbots, agentes conversacionales, interfaces conversacionales, agentes chat y otros términos.

Tipos de bots

Existen diferentes tipos de bots según lo explica Shevat (2017). Entre estos tipos tenemos:

Bots para consumidores

Son bots construidos con propósitos de entretenimiento, mantener al día con noticias, mejorar la productividad y el bienestar de las personas, ayudar a mantenerse en forma, etc

Bots para empresas

Son bots con el propósito de hacer eficiente la ejecución de tareas y procesos de negocios.

Bots de texto

Los bots de texto son bots que soportan una conversación a través de texto. Estos bots suelen implementarse en plataformas como Slack, Facebook, Telegram y WeChat.

Bots de voz

Son bots que se activan a través de comandos de voz. Como ejemplo están: Alexa de Amazon, Cortana de Microsoft, Siri de Apple y Google Assistant.

Trabajo remoto.

Según indica el Ministerio de Trabajo y Promoción del Empleo en su Guía para la aplicación del trabajo remoto, el trabajo remoto es la prestación de servicios de un empleado ubicado en su residencia o desde donde realice el aislamiento social.

El uso de equipos electrónicos, de telecomunicaciones y de servicios de internet, telefonía y otros, permiten hacer trabajo remoto. Todo medio que haga posible la ejecución de tareas fuera de un centro de labores también es una herramienta para el trabajo remoto.

3.2.5. Implementación de las áreas, procesos, sistemas y buenas practicas

Primera parte: Bot de registro de asistencia.

La primera parte del proyecto consistió en la creación del bot con la funcionalidad de registro y visualización de asistencias. Esto se dividió en las siguientes etapas:

Recopilación y definición de requisitos

En esta etapa se tuvieron reuniones virtuales entre el equipo del proyecto para exponer la situación problemática y para asignar funciones a los miembros del equipo. También se tuvo reuniones con el área de RRHH para recoger información del proceso de registro de asistencia y poder saber los requisitos que debía cumplir el bot. Además, hubo reuniones con el área de Business Analytics, donde se conoció las especificaciones que tenía que haber en el bot para que pudiera trabajar correctamente con el sistema de registro de asistencia.

Luego de haber recopilado los requisitos de los interesados, se procedió a analizar qué requisitos se tomarían en cuenta para el bot y pasarían a ser definidos. En la tabla 3 se introduce la lista de requisitos que se consideraron para la primera parte del proyecto.

Tabla 3
Lista de requisitos de la primera parte del proyecto

Código	Requisitos
REQ1	El bot debe permitir que los usuarios registren sus horas de inicio, fin y de refrigerios.
REQ2	El bot debe notificar al usuario inmediatamente sobre el tipo de marcación que ha realizado.
REQ3	El bot debe notificar a todo el equipo sobre las marcaciones que vaya realizando cada miembro del equipo.
REQ4	El bot debe permitir a ciertos usuarios la consulta de la asistencia de uno o más usuarios en un rango de fechas.

Nota. Fuente: Elaboración Propia

Teniendo identificados los requisitos para la primera parte del desarrollo del bot, se procedió a elaborar el cronograma de actividades del proyecto. El cronograma de actividades para la primera parte del proyecto se ubica en el anexo 1.

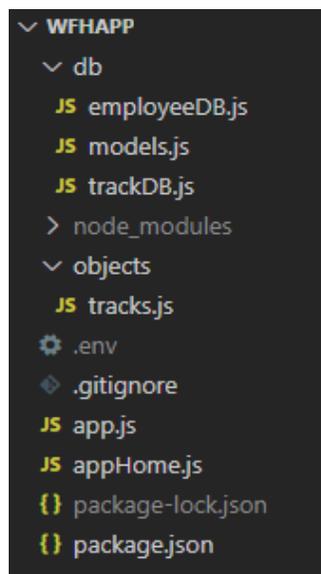
Diseño de la estructura del código, de las interfaces y de los modelos del bot

El diseño de la estructura del código del bot se basó en una aplicación de ejemplo construida por Slack que está disponible públicamente en Github. A continuación, se describen los folders y archivos más importantes de la estructura del código y se adjunta una captura de la estructura completa en la figura 4.

- **/db:** El folder /db contiene la configuración necesaria para poder conectarse a la base de datos. Aquí también se encuentran declarados los esquemas que definirán los documentos (tablas) que utilizara la aplicación.
- **/objects:** Este folder sirve para encapsular a través de métodos las operaciones más comunes que realizaremos en la base de datos.
- **app.js:** Es el archivo que correremos para iniciar la aplicación y es donde la mayor parte de la lógica del bot se encuentra.

Figura 4

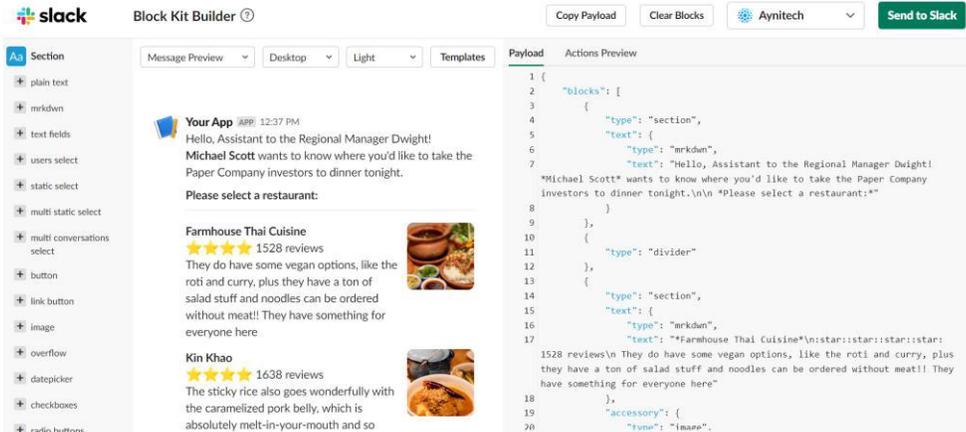
Estructura del código para la primera parte del bot



Nota. Fuente: Elaboración propia.

En esta etapa también se diseñaron los prototipos para la interfaz gráfica del bot. Los prototipos fueron creados utilizando Block Kit Builder, una herramienta de Slack para la construcción rápida de prototipos, la cual puede observarse en la figura 5.

Figura 5
Herramienta de Slack para la construcción de prototipos: Block Kit Builder



Nota. Fuente: Elaboración propia.

Los siguientes prototipos se desarrollaron en la primera parte del proyecto con la herramienta Block Kit Builder y se adjunta los prototipos en formato JSON generados en el anexo 3:

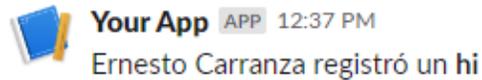
- **Prototipo de formulario para la marcación de asistencia:** Este prototipo es una previsualización del formulario que permitirá a los usuarios realizar una marcación a través de la selección en un menú con una lista de opciones. A continuación, se puede ver en la figura 6 como quedó el prototipo para esta interfaz.

Figura 6
Prototipo de formulario para la marcación de asistencia

Nota: Fuente: Elaboración propia.

- **Prototipo de notificación de asistencia:** En este prototipo se puede apreciar cómo el bot mostrará la marcación realizada por cada usuario. Este prototipo ha sido incluido en la figura 7.

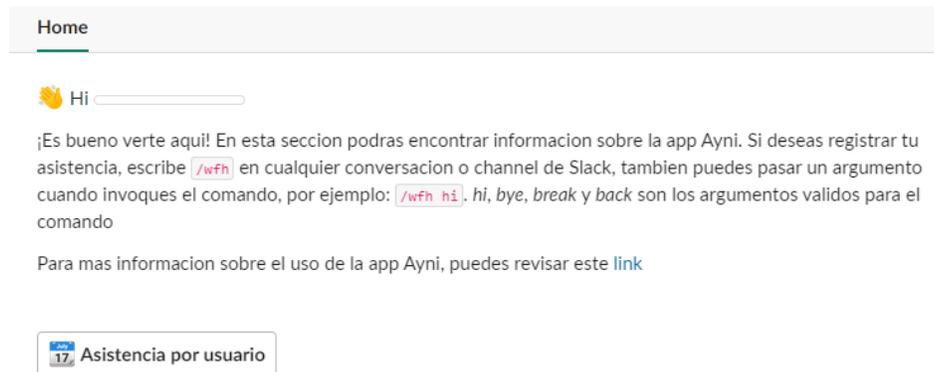
Figura 7
Prototipo de notificación de asistencia



Nota: Fuente: Elaboración propia.

- **Prototipo del App Home con funciones para RRHH:** El prototipo de la pestaña Home contara con un mensaje de bienvenida y un resumen de algunas funciones del bot. Además, incluirá un botón que solo RRHH y los admins podrán ver. El botón permitirá abrir un formulario para la consulta de asistencia de los empleados. El prototipo del Home propuesto puede apreciarse en la figura 8.

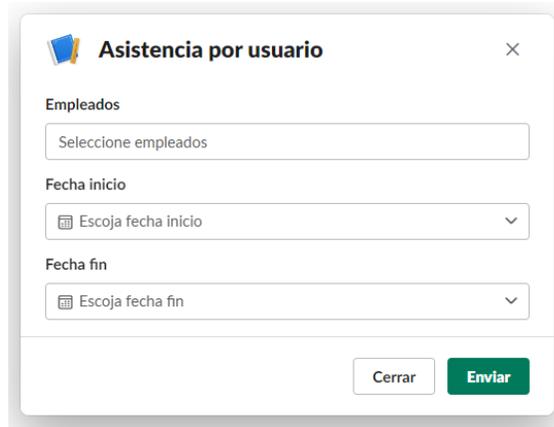
Figura 8
Prototipo del Home con funciones para RRHH



Nota. Fuente: Elaboración propia.

- **Prototipo del formulario de consulta de asistencia por fechas:** Este prototipo contara con un menú de selección para escoger uno o varios empleados de una lista y con campos para elegir fechas de inicio y fin. La figura 9 expone el prototipo para esta interfaz.

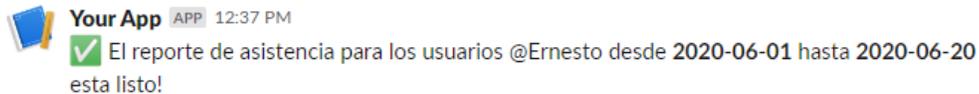
Figura 9
Prototipo del formulario de consulta de asistencia por fechas



Nota. Fuente: Elaboración propia.

- **Prototipo de notificación de consulta de asistencia:** En este prototipo se incluye el mensaje que recibirán los usuarios que utilicen la función de consulta de asistencia. Nótese como el prototipo muestra el mensaje en la figura 10.

Figura 10
Prototipo de notificación de consulta de asistencia



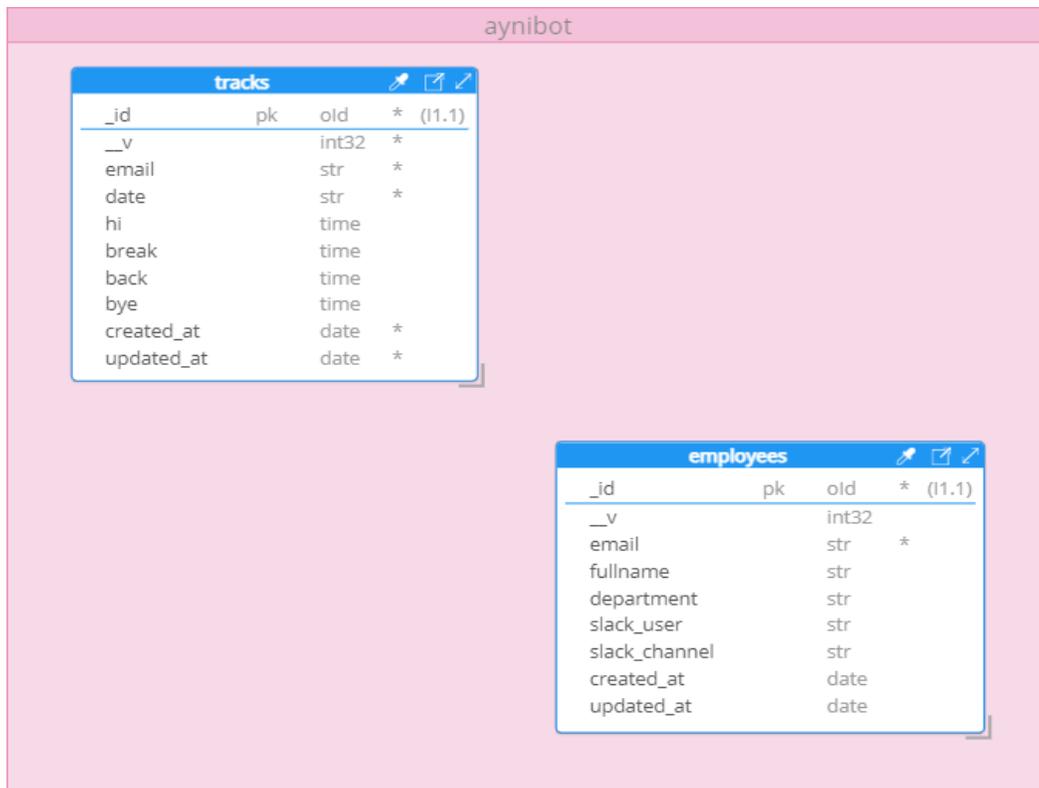
Nota. Fuente: Elaboración propia.

Parte de la etapa de diseño incluyo el modelado de la data necesaria para el funcionamiento de la aplicación. Es así que se creó dos modelos:

- **Modelo Employees:** Representa a los usuarios del sistema y tiene como atributos requeridos: email del usuario, ID del usuario en Slack, e ID del canal de Slack del cual el usuario es miembro.
- **Modelo Tracks:** Un modelo que representa el seguimiento a las horas de la jornada de un empleado en una fecha específica. Los atributos requeridos de este modelo son: email del usuario al que se le realiza el seguimiento, fecha del seguimiento, hora de inicio de las labores (hi), hora de fin de las labores (bye), hora de inicio de refrigerio (break) y hora de fin de refrigerio (back).

Dado que para el bot se ideó usar una base de datos NoSQL como lo es MongoDB, los modelos no están forzados a tener una relación entre ellos y esto se puede apreciar en el diagrama de datos que se muestra en la figura 11.

Figura 11
Diagrama de datos



Nota. Fuente: Elaboración propia

Desarrollo del bot

La etapa de desarrollo se inició con la selección de las tecnologías que se utilizarían para la construcción del Slack bot.

Selección de tecnologías

Debido a la experiencia previa del desarrollador, los únicos lenguajes de programación que se consideraron como posibles opciones fueron JavaScript y Python. Un aspecto que fue clave para la elección del lenguaje de programación es el hecho que de antemano ya se tenía pensado utilizar el framework Bolt, el cual se hallaba disponible para Python solo en su versión beta. Esto hizo la decisión de usar JavaScript más sencilla.

Resuelto el tema de que lenguaje de programación se usaría, la selección de las demás herramientas se basó en el hecho de que tecnologías tenían mejor interoperabilidad con el framework y lenguaje elegidos.

La elección del tipo de base de datos y de la base de datos en si fue influenciada mayormente porque se preveía que la información tenía una estructura sencilla y no iba a ser necesario modelar un gran número de entidades.

Para la creación del Slack bot se utilizaron las herramientas y tecnologías listadas en la Tabla 4.

Tabla 4
Tecnologías y herramientas utilizadas en el desarrollo del bot

Tecnologías y Herramientas	
Desarrollo	Producción
<p>Lenguaje y entorno de ejecución:</p> <ul style="list-style-type: none"> ▪ JavaScript ▪ Node.js v12 <p>Base de datos:</p> <ul style="list-style-type: none"> ▪ MongoDB v4.2.19 <p>Librerías:</p> <ul style="list-style-type: none"> ▪ Bolt para JavaScript v2.7.0 ▪ Mongoose v5.13.2 ▪ Xlsx v0.16.2 <p>Herramientas:</p> <ul style="list-style-type: none"> ▪ MongoDB Compass ▪ Visual Studio Code ▪ Ngrok ▪ Git ▪ Github 	<p>Librerías:</p> <ul style="list-style-type: none"> ▪ Pm2 v4.4.0 <p>Herramientas:</p> <ul style="list-style-type: none"> ▪ Nginx v1.14.1

Nota. Fuente: Elaboración propia

Creación y configuración del Slack bot

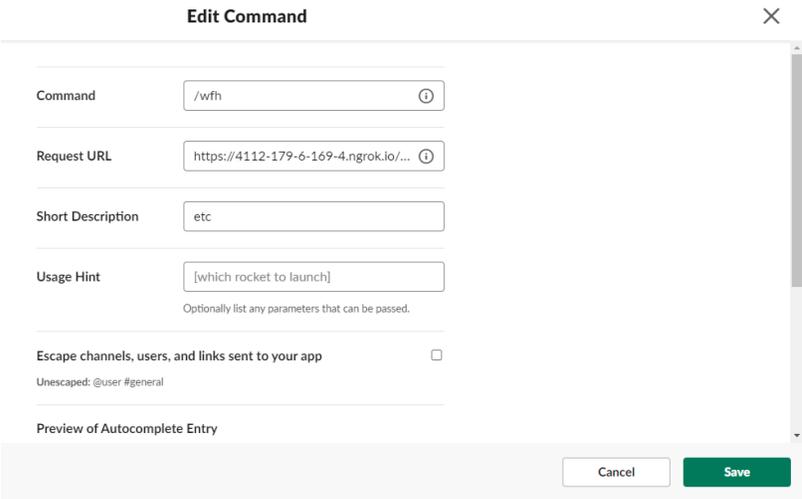
La siguiente tarea de la etapa de desarrollo fue la creación del bot en un Workspace exclusivo para el proceso de desarrollo pues así se garantizaba que no se impactarían las actividades realizadas en el Workspace utilizado por la empresa.

Una vez creada la app en el Workspace para desarrollo, se procedió a la configuración de la app. Primero se añadió la funcionalidad de Slash Commands, a través de la creación de un comando al cual se llamó wfh, dicho nombre fue escogido por ser una abreviación de la palabra working from home.

Uno de los requisitos para la creación de este tipo de comandos, es que hay que proveerle un endpoint al que Slack pueda enviar peticiones cada vez que un usuario invoque el comando.

La figura 12 brinda una visualización de cómo luce el comando wfh luego de haber sido configurado.

Figura 12
Creación del Slash Command: /wfh



The image shows a 'Edit Command' dialog box with the following fields and values:

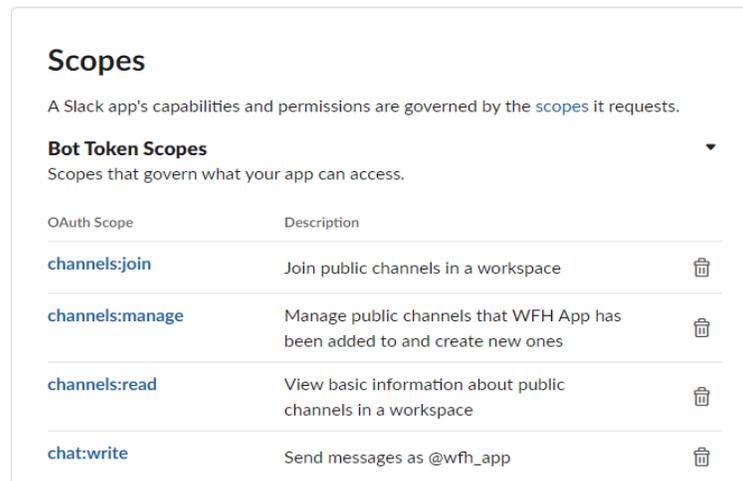
- Command: /wfh
- Request URL: https://4112-179-6-169-4.ngrok.io/...
- Short Description: etc
- Usage Hint: [which rocket to launch]
- Escape channels, users, and links sent to your app:

Below the checkbox, it says 'Unescaped: @user #general'. At the bottom, there is a 'Preview of Autocomplete Entry' section and two buttons: 'Cancel' and 'Save'.

Nota. Fuente: Elaboración propia

El siguiente paso de la configuración de la app, fue la asignación de Scopes para dar a la app permiso para realizar ciertas acciones. Algunos de los Scopes necesarios para el funcionamiento del bot se aprecian en la figura 13.

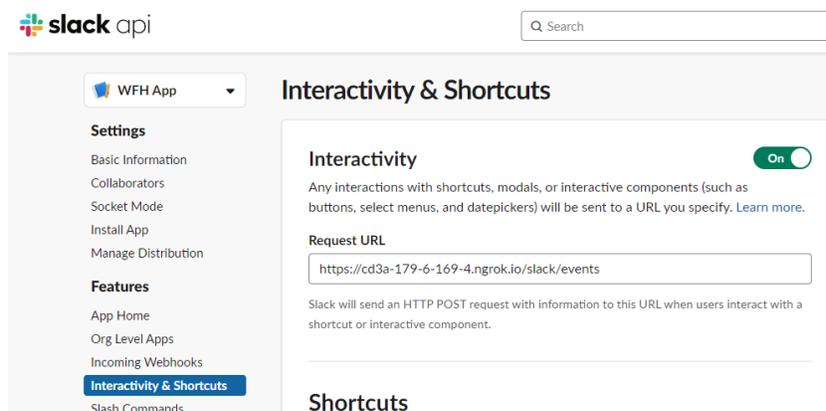
Figura 13
Menú para la asignación de Scopes



Nota. Fuente: Elaboración propia

Finalmente, se tenía que habilitar la funcionalidad de Interactivity para permitir que los componentes interactivos de las interfaces como botones, menús de selección y datepickers trabajen adecuadamente. Al igual que con los Slash Commands, la funcionalidad de Interactivity también requiere de un endpoint para poder habilitarse. En la figura 14 puede verse que ya se tiene la Interactivity activada y que se configuró un endpoint para recibir las peticiones de Slack.

Figura 14
Configuración de Interactivity en el bot



Nota. Fuente: Elaboración propia

Una vez concluida la configuración, ya estábamos listos para instalar el bot en el workspace para pruebas y empezar con la codificación de las funcionalidades.

Comando /wfh

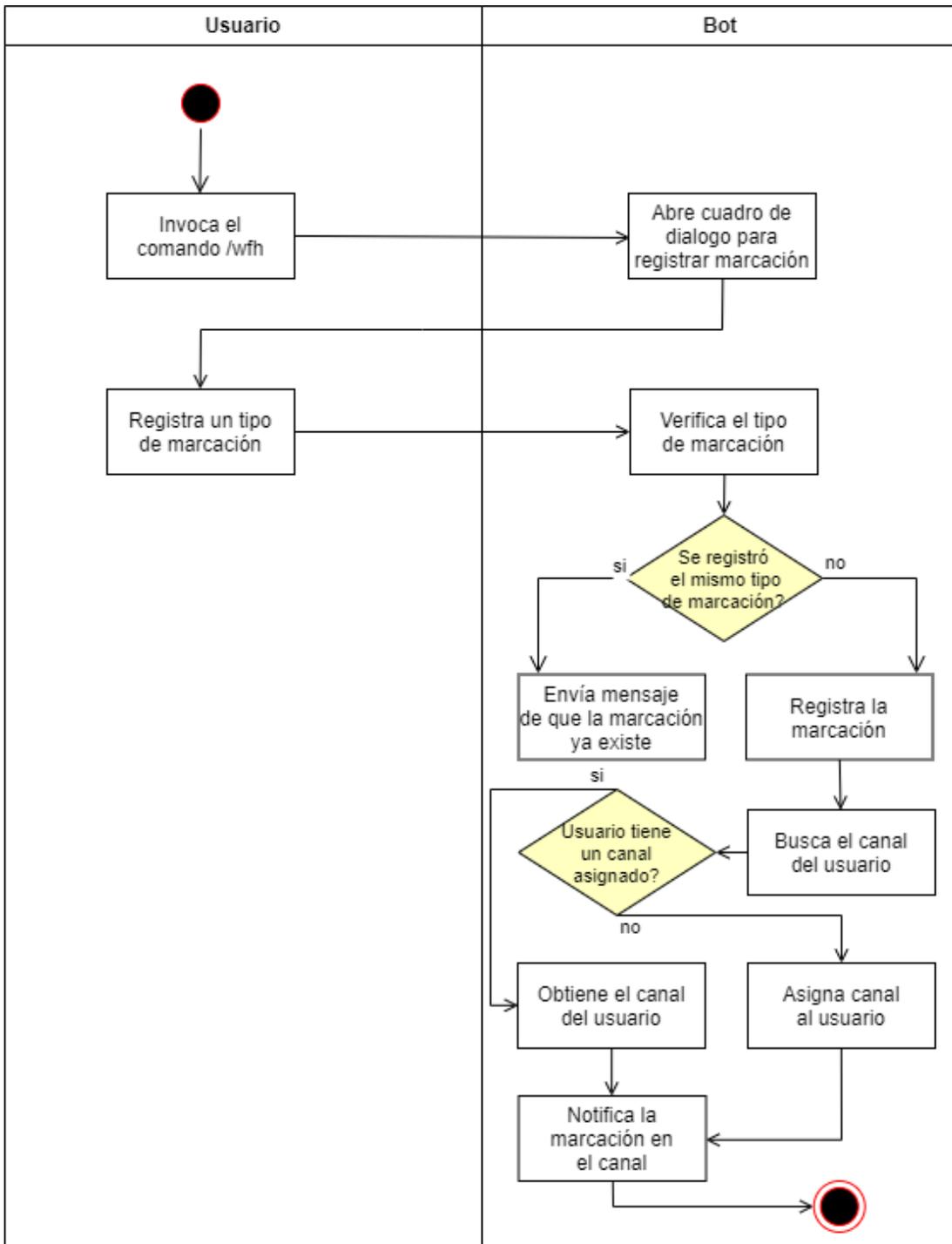
Los usuarios interactuaran con Slack para registrar su asistencia mediante el uso del comando wfh, se decidió que este registro sea a través de un Slash Command pues su utilización es sencilla, son relativamente fáciles de implementar y pueden activarse desde muchos lugares dentro de Slack. Además, usar un Slash Command para esta funcionalidad ofrecía la posibilidad de que el registro de asistencias se realice en el menor número de pasos posible, que es algo que se busca en el tipo de aplicación que se quería implementar ya que el registro de tiempos debe ser inmediato y preciso.

La forma en la que funciona el comando es como sigue:

- El usuario invoca el comando al escribir /wfh (nótese la barra diagonal) en el campo para ingreso de mensajes que tiene Slack y presiona enter.
- El bot responde mostrando un cuadro de dialogo (Modal) que contiene las opciones validas de marcación: hi, break, back y bye.
- El usuario escoge una de las opciones mencionadas y envía esta información para su procesamiento al presionar un botón.
- La marcación escogida es evaluada y si el mismo tipo de marcación no ha sido registrado en la fecha actual, entonces se persiste y se continua con el siguiente paso. De ya encontrarse registro de una marcación del mismo tipo, el bot notificara que ese tipo de marcación ya fue realizado a determinada hora.
- Para enviar un mensaje que informe sobre la marcación al canal donde se encuentran los compañeros de equipo del usuario, es necesario saber que canal pertenece. Este canal que el usuario tiene asignado se ha almacenado la primera vez que utilizo el comando y puede obtenerse de la base de datos.
- El bot notifica la marcación en el canal del usuario y también a él o ella personalmente.

Los pasos descritos líneas arriba también pueden verse en el diagrama de flujo de la figura 15.

Figura 15
 Diagrama de flujo del funcionamiento del comando wfh



Nota. Fuente: Elaboración Propia

App home con funcionalidad para recursos humanos

Durante la etapa de mantenimiento se agregó una nueva funcionalidad al bot para que se pueda obtener reportes de asistencia a través de Slack. A diferencia del comando /wfh que estaba disponible para todos los usuarios, solo el área de recursos humanos y los administradores de Slack en la empresa debían tener acceso a esta función. Es con esta premisa que se evaluó con que componente de Slack se podía limitar el acceso a la funcionalidad que se quería implementar. Luego de una examinación de los tipos de interfaces disponibles en ese momento, se concluyó que una buena alternativa sería utilizar el Home de la app ya que esta interfaz brindaba la capacidad de personalizar lo que cada usuario veía en ella y esto hacía posible que pudiéramos mostrar solo a usuarios específicos la función de obtención de reportes.

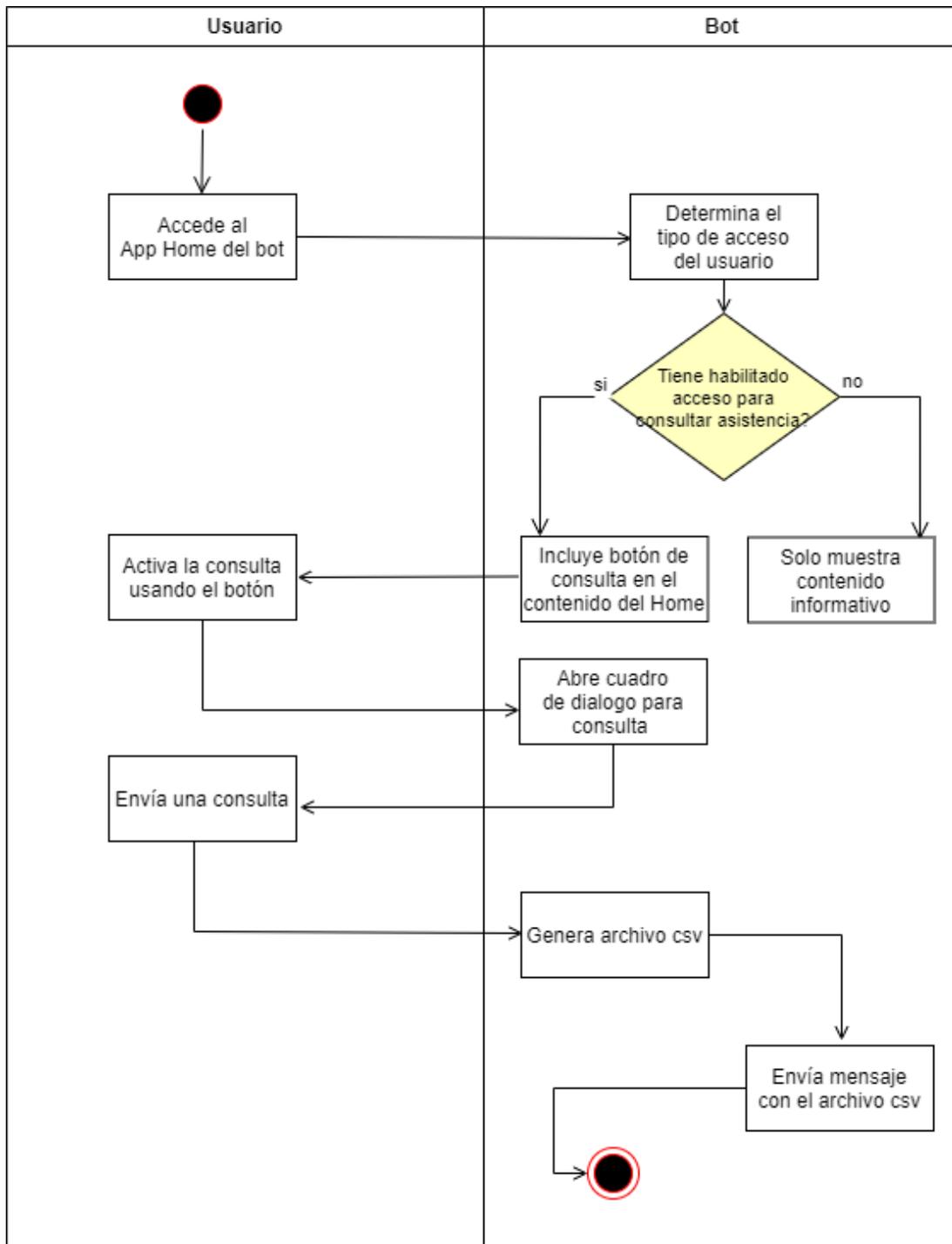
Es así que se activó el App home del bot y se incluyó un mensaje de bienvenida con instrucciones de cómo utilizar el bot, esto podía ser visto por todos, pero solo los usuarios designados podían además ver un botón incluido en la interfaz. Es mediante este botón que se podía acceder a la consulta de asistencia de todos los empleados.

A continuación, describimos los pasos que se debían seguir para realizar la consulta de asistencia:

- El usuario selecciona el bot y se ubica en la pestaña Home de la app.
- Se muestra el contenido del Home a todos los usuarios y se determina si el usuario tiene permitido visualizar el botón para la consulta de asistencia. De estar habilitado para ello, también se incluye este botón.
- El usuario utiliza el botón y esto abre un cuadro de dialogo donde se puede escoger uno o varios empleados y un rango de fechas.
- El usuario llena los inputs con los valores deseados y envía su consulta al clicar un botón.
- Se realiza el cálculo con la información provista, construyendo un archivo csv con los registros de asistencia.
- El bot envía un mensaje personal al usuario adjuntando el archivo csv.

En la figura 16 se puede contemplar como se da este proceso en un diagrama de flujo.

Figura 16
 Diagrama de flujo del funcionamiento de la consulta de asistencia.



Nota. Fuente: Elaboración Propia

Pruebas del bot

Las pruebas solo se enfocaron en la validar la logica de negocio del bot, pues según señala Risquez (2019) los componentes como la API de Slack que forman parte del bot son mantenidos por terceros. Debido a que la cantidad de horas disponibles por día era muy reducida, se decidió realizar solo un tipo de pruebas. Las pruebas de sistema realizadas al bot fueron las siguientes:

- Se replicaron algunos canales del espacio de trabajo de Aynitech en el espacio de trabajo de pruebas.
- Se añadieron múltiples usuarios al workspace de pruebas.
- Se invitaron a los usuarios de prueba a distintos canales.
- Se utilizó el bot con usuarios diferentes.
- Se validó que las notificaciones se envíen a los canales que les corresponden a los usuarios.
- Se dio acceso a la función de consulta de asistencia a determinados usuarios.
- Se verificó que solo los usuarios con acceso podían ver la opción de consulta.
- Se comprobó que el reporte de consulta de asistencias tuviera la información correcta.

Mantenimiento del bot

En esta etapa se corrigieron algunos errores que se encontraron luego del lanzamiento de la primera versión del bot. Uno de los errores detectados estaba relacionado al uso de un formato de fecha y hora incorrectos, que dificultaba el procesamiento de la información que realizaba el sistema mantenido por el área de Business Analytics. También se hicieron algunas optimizaciones en el código y se elaboró el manual de usuario del bot que se adjunta en el anexo 4.

Segunda parte: Bot para el reporte de tareas.

En la segunda parte se agregó al bot la opción para que los empleados registren las tareas que llevaron a cabo durante el día. Para esta funcionalidad el bot trabajaría con una aplicación web que serviría para gestionar las tareas registradas. La comunicación entre estos dos componentes se realizó a través de una API provisionada por el backend de la aplicación web.

El autor de este trabajo se dedicó exclusivamente a añadir las nuevas funciones requeridas al bot y la parte web del proyecto estuvo a cargo de otro desarrollador que se encargó de la construcción de los lados frontend y backend. También se contó con el apoyo de un team lead para labores de preparación de la infraestructura en la que se desplegaría la web.

Este proyecto comenzó con una reunión inicial donde todos los miembros del equipo participaron en el modelado de la información y definición de actividades que se llevarían a cabo. El entregable producto de esta reunión fue un cronograma con las tareas a realizar, el cual se incluyó en el anexo 2.

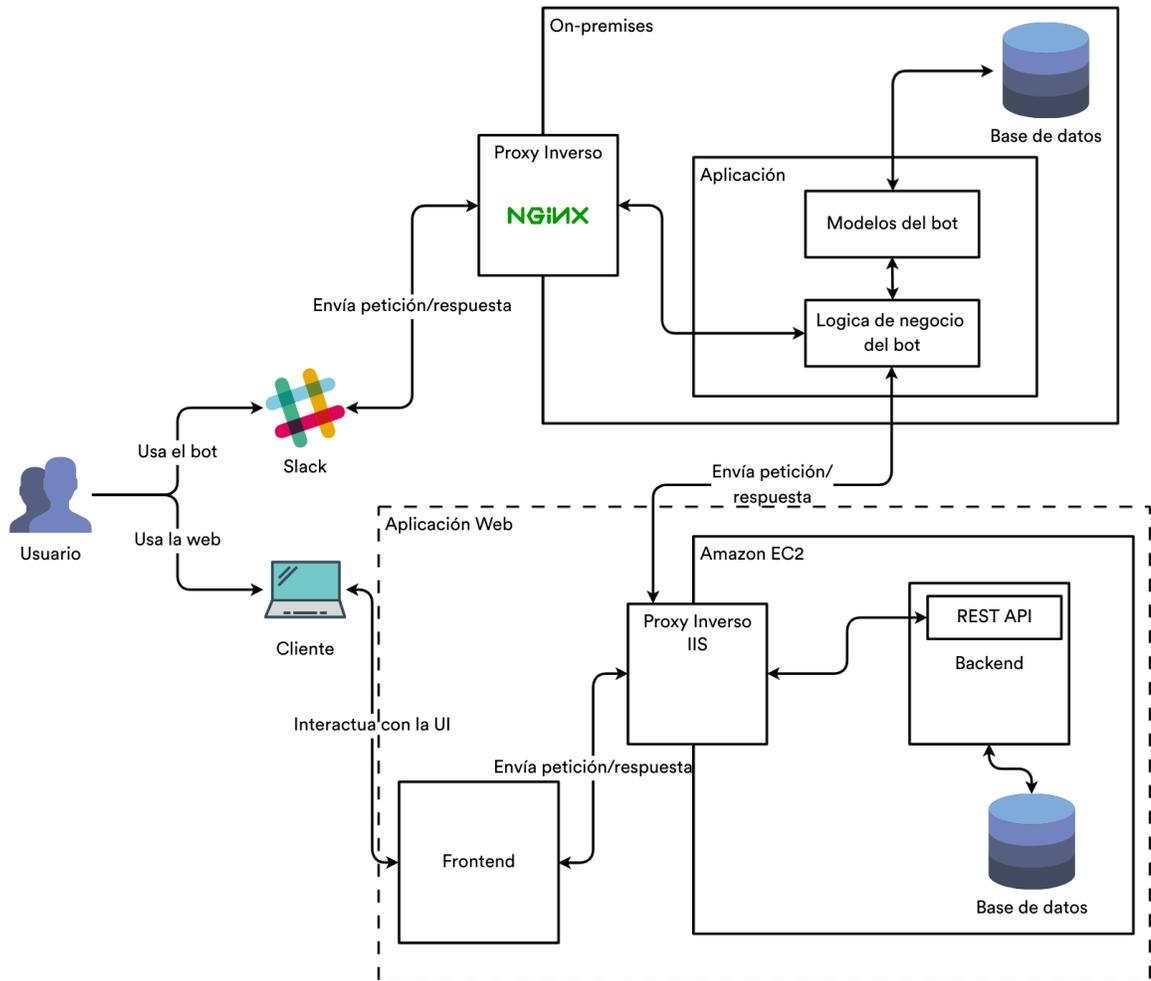
Inicialmente se planeó tener la funcionalidad de reporte de tareas en un nuevo bot pero esto cambio cuando se comprobó que debido a que se usaba el plan gratuito de Slack, el número de bots que podían agregarse al Workspace de la empresa estaba limitado a 10. Es así que se acordó que la funcionalidad debía incluirse en el bot de asistencia, por eso no hubo necesidad de preparar un nuevo entorno donde correr otro bot y de tampoco hacer una selección de tecnologías. Del lado de la web si se tuvo que dedicar esfuerzo para acondicionar un ambiente para que se pueda realizar el lanzamiento. En la tabla 5 se puede ver que tecnologías se usaron para la implementación de la aplicación web.

Tabla 5
Tecnologías y herramientas utilizadas en la construcción del componente web

Tecnologías y Herramientas	
Backend	Frontend
Lenguaje: <ul style="list-style-type: none"> ▪ JavaScript ▪ Node.js Base de datos: <ul style="list-style-type: none"> ▪ MySQL Librerías: <ul style="list-style-type: none"> ▪ Express.js ▪ Sequelize Herramientas: <ul style="list-style-type: none"> ▪ IIS ▪ MySQL Workbench ▪ Postman 	Lenguaje: <ul style="list-style-type: none"> ▪ JavaScript Librerías: <ul style="list-style-type: none"> ▪ Vue.js ▪ Vuetify

El bot y la aplicación web en conjunto forman un sistema que le permite a la empresa tener visibilidad de las tareas en las que se encuentran enfocados sus colaboradores. En la figura 18 se expone la arquitectura de estos dos componentes.

Figura 18
Diagrama de arquitectura del bot y la aplicación web



Nota. Fuente: Elaboración propia

3.3. Evaluación

3.3.1. Evaluación Económica/ Evaluación Costo-Beneficio

Debido a que para la construcción del bot se utilizó mayormente tecnologías open source y a que el despliegue se realizó en un servidor interno de la compañía, el costo del proyecto está basado en las remuneraciones que se dieron al equipo. Para obtener el presupuesto utilizado en la primera fase, se consideró el costo de las horas que el personal dedico al desarrollo que tuvo una duración de 1 mes. Este costo fue incluido en el salario de los involucrados como una bonificación por horas extras al final del mes. A continuación, se muestra en la tabla 6 como se realizó el cálculo de este costo.

Tabla 6
Costo del desarrollo del bot de registro de asistencia

Personal	Costo por hora	Horas dedicadas	Costo subtotal
Team Lead	S/ 20.00	20	S/ 400.00
Developer	S/ 10.00	40	S/ 400.00
Costo total			S/ 800.00

Nota. Fuente: Elaboración propia

En la segunda fase del proyecto, los miembros del equipo también trabajaron horas extra fuera de su horario habitual por un periodo de 2 semanas. El cálculo del costo total se detalla en la tabla 7 de la misma manera que se hizo para la primera fase de desarrollo.

Tabla 7
Costo del desarrollo del bot y de la aplicación web para la segunda parte

Personal	Costo por hora	Horas dedicadas	Costo subtotal
Team Lead	S/ 20.00	20	S/ 400.00
Bot Developer	S/ 10.00	40	S/ 400.00
Web Developer	S/ 10.00	40	S/ 400.00
Costo total			S/ 1200.00

Nota. Fuente: Elaboración propia

Antes del lanzamiento del bot de registro de asistencias, se realizaba una actividad de revisión manual de las asistencias que tuvo inicio el 14 de marzo de 2020 y continuo hasta el 5 de mayo de 2020. Esta actividad estuvo a cargo de un analista del área de recursos humanos que empleaba 3 horas diarias para la revisión y recopilación de las asistencias en una hoja Excel. En la tabla 8, se calcula el costo en el que se incurría al no tener automatizado este proceso.

Tabla 8
Costo del registro de asistencia antes de la implementación del bot

Personal	Costo por hora	Cantidad de horas/día	Número de días/semana	Costo semanal	Costo mensual
Analista de RRHH	S/ 7.00	3	5	S/ 105.00	S/ 420.00

Nota. Fuente: Elaboración propia

Con los costos identificados, se calculó el flujo de caja para los 6 primeros meses posteriores a la implementación del bot (mayo de 2020 hasta octubre 2020) como puede verse en la tabla 9.

Tabla 9
Flujo de caja

	Mayo de 2020	Junio de 2020	Julio de 2020	Agosto de 2020	Setiembre de 2020	Octubre de 2020
Costo	S/ 800.00	S/ 0.00	S/ 0.00	S/ 0.00	S/ 0.00	S/ 0.00
Beneficio	S/ 0.00	S/ 420.00	S/ 420.00	S/ 420.00	S/ 420.00	S/ 420.00
Beneficio Neto	-S/800.00	S/ 420.00	S/ 420.00	S/ 420.00	S/ 420.00	S/ 420.00

Nota. Fuente: Elaboración propia

3.3.2. Interpretación Del VAN y Del TIR

Se realizó el cálculo del VAN y del TIR usando una tasa de descuento del 12%. Los resultados obtenidos se muestran en la tabla 10.

Tabla 10
VAN y TIR al sexto mes

Formula	Monto
VAN al sexto mes	S/ 1075.00
TIR al sexto mes	44%

Nota. Fuente: Elaboración propia

El VAN conseguido es S/ 1075.00, lo que indica que la inversión que hizo la empresa en el proyecto empezó a generar beneficios a partir del sexto mes, es decir octubre 2020. De igual manera, al tener un TIR de 44% se estima que luego de implementado el bot, a la empresa le tomó 6 meses para recuperar su inversión.

Al analizar los dos montos obtenidos podemos concluir que el proyecto fue viable para la empresa pues se puede apreciar que se obtuvo una rentabilidad en un tiempo relativamente corto.

CAPÍTULO IV REFLEXIÓN CRÍTICA DE LA EXPERIENCIA

A lo largo de la experiencia sobre la que trata este trabajo, el autor fue el principal encargado de la construcción del bot. Este trabajo permitió al autor plasmar todo lo que había aprendido trabajando en el desarrollo de Slack bots para el cliente externo en USA y trasladarlo hacia las necesidades de Aynitech.

Se logró aprovechar las bondades que ofrecía Slack para dar solución a un problema de la empresa y esto se hizo ahorrando costos pues no se necesitó recurrir a algún plan de pago de Slack.

La decisión de usar MongoDB fue un acierto pues tiempo después hubo requerimientos para agregar más funciones al bot. Mucha de esta nueva funcionalidad demandaba la modificación de los esquemas de los modelos definidos al inicio, hacer este cambio tiene una complejidad considerable en aplicativos que usan sistemas de base de datos relacionales y que ya se encuentran en producción, pero en nuestro caso, esta tarea resulto ser simple de ejecutar gracias a la flexibilidad del sistema de base de datos escogido.

Algunos errores surgidos luego del lanzamiento del bot como, por ejemplo: los ocasionados por el uso de un formato incorrecto de fecha y hora pudieron evitarse si se hubiera contado con pruebas unitarias.

Si bien el desarrollo del bot se presentaba como un proyecto pequeño, se debió utilizar una metodología más elaborada.

CAPÍTULO V CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

1. Se desarrolló un Slack bot que mejoró el control del trabajo remoto en la empresa. El bot implementado colaboró en la adopción del teletrabajo en Aynitech.
2. Se construyó un bot con funcionalidad para el registro de las horas de entrada, de salida y de refrigerios de los empleados. Además, se incluyó en el bot la opción para consultar las horas fichadas por los empleados en determinadas fechas, esta función se habilitó solo a cierto personal de áreas como RRHH y Soporte de TI. El beneficio más evidente que se consiguió con estas capacidades fue la eliminación del trabajo y esfuerzo que tenía que hacer RRHH para recolectar manualmente las horas de todos los empleados.
3. Se incorporó al bot la funcionalidad de reporte de tareas diarias para los empleados. El bot hace él envió de la información recogida a la API de una aplicación web desarrollada para gestionar las tareas reportadas. Esta función del bot más la app web, permiten a la empresa tener una opción disponible para llevar a cabo el seguimiento a las tareas y actividades que efectúan sus trabajadores. La empresa ahora puede proveer de reportes de actividades a sus clientes y también puede detectar problemas de rendimiento en sus colaboradores.

5.2. Recomendaciones

Debido a que la aplicación solo funciona cuando los usuarios ejecutan alguna acción, no es muy necesario tener el bot corriendo todo el tiempo como un servicio y una opción a considerar para ahorrar costos podría ser adoptar un modelo Serverless. De este modo solo se ejecutarían partes de la aplicación según se disparen eventos por las acciones de los usuarios.

Conforme se añadan nuevas funciones al bot y este se haga más complejo, será más difícil tener organizado la base de código del proyecto. Es por ello que se recomienda reemplazar JavaScript por TypeScript, que debido a ser un lenguaje de tipado estático facilita la detección de errores y la claridad del código y estos son solo algunos beneficios que lo hacen ideal para proyectos más grandes.

Otra posible mejora que puede aplicarse para futuros proyectos relacionados al bot, es usar una metodología de desarrollo más establecida y que sea de conocida por el equipo de proyecto.

5.3. Fuentes De Información

- Aynitech. (2022). About Us. Recuperado de <https://www.aynitech.com>
- Aynitech. (2022). Our Companies. Recuperado de <https://www.aynitech.com>
- Guía para la aplicación del trabajo remoto. (Marzo, 2020). *Ministerio de Trabajo y Promoción del Empleo*. Recuperado de https://cdn.www.gob.pe/uploads/document/file/662676/GUIA_PARA_LA_APLICACION_DEL_TRABAJO_REMOTO.pdf
- Markovich, D. (2021). *Supercharge Your Slack Productivity*. (1.^a ed.). Birmingham, UK: Packt Publishing Ltd.
- Monjardín López, E. D. (2018). *HR Bot: robot para la gestión de recursos humanos (Proyecto)*. Universidad ORT Uruguay, Facultad de Ingeniería. Recuperado de <https://dspace.ort.edu.uy/handle/20.500.11968/3862>
- Rísquez Almodóvar, P. (2019). *Desarrollo de un chatbot para modelado colaborativo en Slack*. Universidad Autónoma de Madrid, Departamento de Ingeniería Informática. Recuperado de <https://repositorio.uam.es/handle/10486/688978>
- Shevat, A. (2017). *Designing Bots*. (1.^a ed.). United States of America: O'Reilly Media, Inc.
- Slack. (2022). Precios. Recuperado de <https://slack.com/intl/es-pe/pricing>
- Slack. (2022). ¿Qué es Slack? Recuperado de <https://slack.com/intl/es-pe/help/articles/115004071768-¿Qué-es-Slack->
- Slack. (2022). Using Slack APIs. Recuperado de <https://api.slack.com/apis>

5.4. Glosario

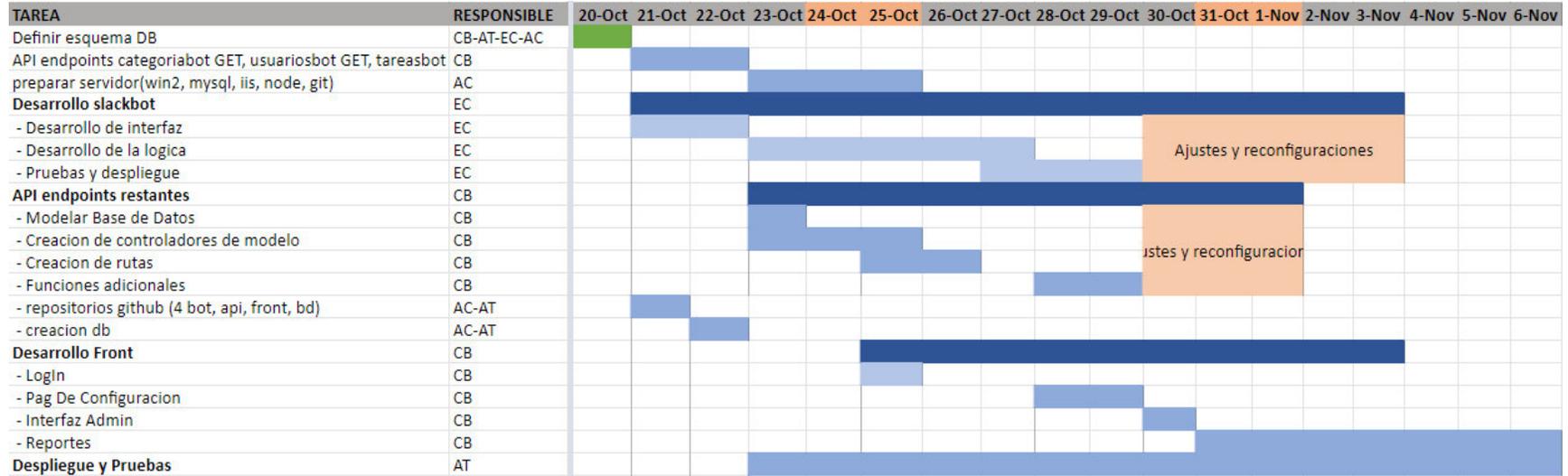
- **App Home:** La pestaña Home es una interfaz dinámica para apps. El App Home puede ser habilitado para cualquier app y se puede acceder a él haciendo click en el nombre de la app. Además, el App Home puede personalizarse para cada usuario que accede a él para mostrar contenido dinámico en una interfaz persistente. Un uso que se suele dar al App Home es para la inclusión de dashboards.
- **BPO:** Son las siglas de Business Process Outsourcing, que es la tercerización de procesos de negocio principalmente del área de TI mediante un proveedor.
- **Channel:** Es el lugar donde Slack organiza las conversaciones. En un canal se pueden compartir mensajes, enlaces, herramientas y otras cosas más. Los canales se suelen crear para cada proyecto, equipo, producto u otro tipo de concepto.
- **Slash Commands:** Son una funcionalidad que se puede añadir a las Slack apps. Estos comandos permiten que los usuarios puedan invocar apps al escribir el nombre del comando en una caja para el envío de mensajes.
- **Workspace:** Un workspace o espacio de trabajo es el centro de comunicaciones que está conformado por canales y es donde los miembros del equipo pueden conectarse y colaborar.

Anexo 2: Cronograma de tareas del bot de reporte de tareas y la aplicación web de gestión de tareas

El siguiente cronograma se creó para el desarrollo de la nueva funcionalidad del bot y la construcción de la parte web:

Figura 20

Cronograma de tareas de la segunda parte del proyecto



Nota. Fuente: Elaborado por el equipo del proyecto

Anexo 3: Prototipos construidos utilizando Block Kit Builder

Los prototipos que se crearon con la herramienta Block Kit Builder, generaron un código en formato JSON que pudo ser utilizado por la aplicación. A continuación, se muestran los payloads de las vistas en formato JSON.

Figura 21
Payload en formato JSON de la interfaz de marcación de asistencia

```
1 {
2   "type": "modal",
3   "title": {
4     "type": "plain_text",
5     "text": "Registro de asistencia",
6     "emoji": true
7   },
8   "submit": {
9     "type": "plain_text",
10    "text": "Enviar",
11    "emoji": true
12  },
13  "close": {
14    "type": "plain_text",
15    "text": "Cerrar",
16    "emoji": true
17  },
18  "blocks": [
19    {
20      "type": "input",
21      "element": {
22        "type": "static_select",
23        "placeholder": {
24          "type": "plain_text",
25          "text": "Elija una opción",
26          "emoji": true
27        },
28        "options": [
29          {
30            "text": {
31              "type": "plain_text",
32              "text": "hi",
33              "emoji": true
34            },
35            "value": "hi"
36          },
37          {
38            "text": {
39              "type": "plain_text",
40              "text": "break",
41              "emoji": true
42            },
43            "value": "break"
44          },
45          {
46            "text": {
47              "type": "plain_text",
48              "text": "back",
49              "emoji": true
50            },
51            "value": "back"
52          },
53          {
54            "text": {
55              "type": "plain_text",
56              "text": "bye",
57              "emoji": true
58            },
59            "value": "bye"
60          }
61        ],
62        "action_id": "static_select-action"
63      },
64      "label": {
65        "type": "plain_text",
66        "text": "Marcación",
67        "emoji": true
68      }
69    }
70  ]
71 }
```

Nota. Fuente: Elaboración propia

Anexo 4: Manual de usuario del bot

Durante la etapa de mantenimiento luego del lanzamiento de la primera versión del bot, se elaboró el manual de usuario en un espacio de Confluence creado para este fin. Se incluye una captura de cómo se visualiza este documento.

Figura 22

Manual de usuario del bot



The screenshot shows a Confluence page within a 'SOPORTE' space. The page title is '002 Registro de asistencia por Slack'. It was created by 'Confluence Aynitech' and last updated on 'Oct 20, 2021'. The content explains that to register assistance via Slack, users must enter the command `/wfh` in any Slack channel or app. A code block shows the command `/wfh` with a red underline. Below the code block, it states that this command activates a form with a dropdown menu for recording types like 'hi', 'break', 'back', and 'bye'. The left sidebar shows a navigation menu with categories like 'Overview', 'Blog', 'Pages', and 'MANUALES', where '002 Registro de ...' is selected.

Nota. Fuente: Captura tomada del espacio de Confluence de la empresa