



TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE APIZACO

NOMBRE DEL PROYECTO
RECRUITING CHATBOT

OPCIÓN QUE CORRESPONDA
TITULACIÓN INTEGRAL INFORME DE RESIDENCIA PROFESIONAL

PARA OBTENER EL TÍTULO PROFESIONAL DE:
INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIONES

PRESENTA:
JESSICA MORALES AGUILAR

NOMBRE DEL ASESOR:
DR. PERFECTO MALAQUÍAS QUINTERO FLORES



Educación
Secretaría de Educación Pública



TECNOLÓGICO
NACIONAL DE MÉXICO



Instituto Tecnológico de Apizaco
Sistemas y Computación

Apizaco, Tlaxcala, **30/junio/2025**

Oficio No. S.C./OPV/0266/25

ASUNTO: Liberación de proyecto
para titulación integral

ISRAEL JORGE SÁNCHEZ SÁNCHEZ
JEFE DEL DEPTO. DE DIVISIÓN DE ESTUDIOS PROFESIONALES
P R E S E N T E.

POR ESTE MEDIO LE INFORMO QUE HA SIDO LIBERADO EL SIGUIENTE PROYECTO PARA LA TITULACIÓN INTEGRAL.

NOMBRE DEL EGRESADO:	MORALES AGUILAR JESSICA
CARRERA:	INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN Y COMUNICACIONES
NO. DE CONTROL:	20370963
NOMBRE DEL PROYECTO:	"Recruiting Chatbot"
PRODUCTO U OPCIÓN:	TITULACIÓN INTEGRAL INFORME DE RESIDENCIA PROFESIONAL

AGRADEZCO DE ANTEMANO SU VALIOSO APOYO EN ESTA IMPORTANTE ACTIVIDAD PARA LA FORMACIÓN PROFESIONAL DE NUESTROS EGRESADOS.

ATENTAMENTE

*Excelencia en Educación Tecnológica.
Pensar para Servir, Servir para Triunfar®*

MIQUELINA SÁNCHEZ PULIDO
JEFA DEL DEPTO. DE SISTEMAS Y COMPUTACIÓN

ccp. Coordinación de carrera
Expediente
MSP/mrol*



PERFECTO MALAQUÍAS QUINTERO FLORES
ASESOR INTERNO



2025
Año de
La Mujer
Indígena

Av. Instituto Tecnológico No. 418, San Andrés Ahuashuatepec,
Municipio de Tzompantepec, Tlaxcala, Mex C.P. 90491
Tel. (241) 4172010 Ext. 115, e-mail: sistemas@apizaco.tecnm.mx
tecnm.mx | apizaco.tecnm.mx





Educación
Secretaría de Educación Pública



TECNOLÓGICO
NACIONAL DE MÉXICO



Instituto Tecnológico de Apizaco
División de Estudios Profesionales

Tzompantepec, Tlaxcala **26/septiembre/2025**

MORALES AGUILAR JESSICA
EGRESADO(A) DE LA CARRERA DE
INGENIERÍA EN TECNOLOGÍAS DE LA INFORMACIÓN
Y COMUNICACIONES.
CON NÚMERO DE CONTROL 20370963.
P R E S E N T E.

Por este medio, me permito informar a usted que por aprobación de la comisión revisora asignada para valorar el trabajo que mediante la opción **TITULACIÓN INTEGRAL INFORME DE RESIDENCIA PROFESIONAL** y cuyo tema es: **RECRUITING CHATBOT**, conforme a lo establecido en el procedimiento para la obtención del Título Profesional en el Instituto Tecnológico, la División de Estudios Profesionales a mi cargo le otorga la:

AUTORIZACIÓN DE IMPRESIÓN

Debiendo entregar **2 USB** del mismo, apropiadamente empastados, a la brevedad, para presentar su Acto de Recepción Profesional.

Sin otro particular por el momento me despido.

ATENTAMENTE

Excelencia en Educación Tecnológica®
Pensar para Servir, Servir para Triunfar®

CECILIA YARELLY ARREGUÍN LARA
JEFA DE LA DIVISIÓN DE ESTUDIOS PROFESIONALES
INSTITUTO TECNOLÓGICO DE APIZACO
SECRETARÍA DE EDUCACIÓN PÚBLICA
TECNOLÓGICO NACIONAL
DE MÉXICO
ESTUDIOS PROFESIONALES

ccp. Archivo

CYAL/rrc



SECRETARÍA DE EDUCACIÓN PÚBLICA
TECNOLÓGICO NACIONAL
DE MÉXICO
INSTITUTO TECNOLÓGICO DE APIZACO



2025
Año de
La Mujer
Indígena

Av. Instituto Tecnológico No. 418, San Andrés Ahuashuatepec,
Municipio de Tzompantepec, Tlaxcala, Mex. C.P. 90491 Tel. (241) 4172010
Ext. 304, e-mail: dep_titulacion@apizaco.tecnm.mx | apizaco.tecnm.mx



Agradecimientos

A lo largo de este duro de desarrollo profesional, he tenido la suerte de contar con el apoyo de muchas personas que han contribuido de distintas formas a la culminación de esta residencia, quiero expresar mi más sincero agradecimiento a todas las personas y la empresa que contribuyeron de manera significativa a la realización de este proyecto.

En primer lugar, agradezco al Dr. Perfecto Malaquías Quintero Flores, por su guía, acompañamiento y valiosas observaciones a lo largo del desarrollo de este proyecto. Su experiencia y compromiso fueron fundamental para superar los retos que surgieron durante el proceso. De igual manera, también agradezco al Instituto Tecnológico Nacional de Apizaco, por brindarme la oportunidad de formarme académicamente y facilitarme de los recursos necesarios para la elaboración del proyecto.

A mis compañeros y equipo de trabajo, gracias por su colaboración, ideas y apoyo constante durante los sprints para el desarrollo. Su disposición para compartir conocimientos y trabajar en equipo fue parte fundamental para lograr los objetivos que se nos fueron propuestos y avanzar con éxito. De manera especial quiero agradecer a mis amigos, por estar siempre dispuestos a escucharme, motivarme y acompañarme con palabras de aliento en los momentos difíciles.

A Eduardo, mi compañero de vida, gracias por estar siempre a mi lado con paciencia, amor y sobre todo tu apoyo incondicional. Tu motivación constante, tus palabras de aliento junto con tu presencia han sido mi mayor fortaleza a lo largo de este proceso.

Y por supuesto, a mi familia, por su amor y comprensión, y por creer en mi en cada paso. Su esfuerzo por apoyarme también ha sido parte fundamental para poder seguir adelante con determinación y confianza para concluir con este desarrollo.

Resumen

En el presente proyecto, se tuvo como objetivo rediseñar el estilo visual y funcional de una plataforma de administración para un chatbot de reclutamiento. Este sistema tiene como objetivo principal mejorar la experiencia de los usuarios encargados de gestionar los procesos de selección, integrando elementos de inteligencia artificial para lograr una administración más eficiente.

Entre las actividades realizadas se incluyeron el análisis sobre las historias de usuario previamente establecidas, el análisis de la interfaz actual, así como la propuestas e implementación de mejoras visuales mediante el uso de tecnologías web para el diseño de estilos.

Este proyecto permitió aplicar conocimientos teóricos y prácticos adquiridos dentro del aula, al mismo tiempo, desarrollar nuevas habilidades relacionadas con la experiencia de usuario (UX).

Índice

Agradecimientos.....	2
Resumen.....	3
1. Generalidades del proyecto.....	10
1.1. Introducción	10
1.2. Descripción de la empresa	11
1.3. Descripción del cargo del residente	12
1.4. Problemática por resolver.....	13
1.4.1. Principales problemáticas	13
1.5. Objetivos	13
1.5.1. Objetivo General	13
1.5.2. Objetivos Específicos	14
1.6. Justificación	14
2. Marco Teórico.....	15
2.1. Diseño Centrado en el Usuario.....	15
2.2. Experiencia del usuario (UX)	18
2.3. Diseño Web Adaptable	18
2.3.1. Beneficios del Diseño Web Adaptativo.....	19
2.4. Metodología Scrum	20
2.4.1. Beneficios de la Metodología Scrum	20
2.4.2. Roles de la metodología Scrum	21
2.4.3. Scrum Máster	22
2.4.4. Product Owner.....	22
2.4.5. Artefactos de la Metodología Scrum	22
2.5. Desarrollo Front-End	25
2.5.1. Componentes claves del frontend	25
2.5.2. Herramientas y tecnologías Frontend	27
2.5.3. Tecnologías Utilizadas.....	28
3. Desarrollo	33

3.1.	Procedimiento y descripción de actividades	33
3.1.1.	Instalación de Visual Studio	33
3.1.2.	Instalación de Xampp	36
3.1.3.	Uso de GitHub	40
3.1.4.	Sprint 0. Primera Etapa del Desarrollo	44
3.1.5.	Spring 1. Aplicación de Estilos Corporativos y Ajustes Visuales Iniciales	47
3.1.6.	Spring 2. Renombramiento y Personalización	47
3.1.7.	Spring 3. Optimización de Personalización y Ajustes en la Experiencia de Usuario	50
3.1.8.	Spring 4. Mejoras en la Configuración y Finalización	54
3.1.9.	Spring 5. Implementación de Diseño Responsivo para Dispositivos Móviles	55
4.	Resultados	63
4.1.	Resultados sprint 0	63
4.2.	Resultados del Sprint 1	65
4.3.	Resultados del Sprint 2	65
4.4.	Resultados del Spring 3	67
4.5.	Resultados del Sprint 4	70
4.6.	Resultado del Sprint 5	71
5.	Conclusión	79
5.1.	Experiencia profesional adquirida	79
6.	Competencias Desarrolladas y Aplicadas	80
6.1.	Competencias Especificas	80
6.2.	Competencias genéricas	81
7.	Fuentes de información	82

Índice de ilustraciones

Ilustración 1. Despacho 204 GIINTAPE INNOVAHUE	12
Ilustración 2. Diagrama del proceso de diseño centrado en el usuario	16
Ilustración 3: Metodología Scrum	24
Ilustración 4. Logo de HTML	29
Ilustración 5: Logotipo CSS	30
Ilustración 6: Logotipo JavaScript	31
Ilustración 7. Logotipo Bootstrap	32
Ilustración 8. Sitio oficial para descargar Visual Studio Code	33
Ilustración 9: Carpeta donde se descargará	34
Ilustración 10: Instalación de Visual Studio Code	35
Ilustración 11: Pagina inicial del VS	35
Ilustración 12: Extensión LiveServer	36
Ilustración 13: Sitio oficial de descarga XAMPP	37
Ilustración 14: Configuración de XAMPP	37
Ilustración 15. Selección de carpeta para XAMPP	38
Ilustración 16: Ejecución de XAMPP	39
Ilustración 17: Prueba de localhost	40
Ilustración 18: Inicio de sesión en GitHub	41
Ilustración 19: Invitación al repositorio	41
Ilustración 20: Redirección al repositorio	42
Ilustración 21: Clonación del repositorio	43

Ilustración 22. Creación de la rama	43
Ilustración 23: Visualización de la rama en VS	44
Ilustración 24: Login sin modificaciones	44
Ilustración 25: Implementación del input tipo “email”	45
Ilustración 26: Enlace de “Restablecer contraseña” agregado	45
Ilustración 27: Diseño del modal de restablecimiento	46
Ilustración 28: Código JavaScript para apertura y cierre del modal	46
Ilustración 29: Colores corporativos	47
Ilustración 30: Cambio de nombres en el menú de navegación.....	48
Ilustración 31: Campo color picker.....	48
<i>Ilustración 32: Botón Guardar</i>	<i>49</i>
<i>Ilustración 33: Botón salir</i>	<i>49</i>
<i>Ilustración 34: Estilo de botones.....</i>	<i>49</i>
Ilustración 35: Div de la url.....	50
Ilustración 36: Botón generar	50
<i>Ilustración 37: Código para el input.....</i>	<i>51</i>
<i>Ilustración 38: Código de botón actualizar.....</i>	<i>51</i>
Ilustración 39: Actualización del botón actualizar.....	52
Ilustración 40: Código JavaScript para almacenar.....	52
Ilustración 41: Código JavaScript para recuperar imagen	52
Ilustración 42: Estilos para adecuar el logotipo	53
Ilustración 43: Color picker en burbuja	53

Ilustración 44: Cambio de nombre en la etiqueta	54
Ilustración 45: Eliminación de línea de código	55
Ilustración 46: Cambio a textarea y limitación	55
<i>Ilustración 47: Ajuste del contenedor</i>	<i>56</i>
<i>Ilustración 48: Ajuste del contenedor de los botones</i>	<i>56</i>
<i>Ilustración 49: Adaptación del texto en móviles</i>	<i>57</i>
Ilustración 50: Estilo para la url	57
Ilustración 51: Adaptación del input para dispositivos móviles	58
Ilustración 52: Modificación del Texto	58
Ilustración 53: Código para la resolución 1300 px	59
Ilustración 54: Resolución para tabletas	59
Ilustración 55: Disminución de la fuente para la resolución de celular	59
Ilustración 56: Disminución del icono representativo de la pagina.....	59
Ilustración 57: Reestructuración de los botones	60
Ilustración 58: Código responsivo para tabletas	61
Ilustración 59: Código responsivo para móviles	61
Ilustración 60: Contenedor responsivo para tabletas	62
Ilustración 61: Contenedor responsivo para móviles	62
Ilustración 62: Modificación de fuente	62
Ilustración 63: Login actualizado	64
Ilustración 64: Campo de correo electrónico	64
Ilustración 65: Demostración activa del modal	64

Ilustración 66: Menú actualizado	65
Ilustración 67: Campos color picker en la página de Estilo.....	66
Ilustración 68: Separación de botones “Guardar” y “Salir”	66
Ilustración 69: Sección de URL agregada en la página de Finalizar	67
Ilustración 70: Campo para definir la URL del logo en la página de Estilo	68
Ilustración 71: Mejoras al logotipo	69
Ilustración 72: Visualización del logo en página de Burbuja.....	69
Ilustración 73: Color picker funcional.....	69
Ilustración 74: Cambio de etiquetas en la pantalla mensaje inicial	70
Ilustración 75: Campo de mensaje de despedida con textarea y contador de caracteres	71
Ilustración 76: Modificación de textos descriptivos para mayor claridad	71
Ilustración 77: Vista responsiva de la página de Inicio en celular	72
Ilustración 78: Vista responsiva de la página de Inicio en tabletas	72
Ilustración 79: Vista responsiva de la página de Estilo en dispositivos de tabletas.....	73
Ilustración 80: Vista responsiva de la página de Estilo en dispositivos móviles	74
Ilustración 81: Diseño adaptado de la página de Burbuja en tableta	75
Ilustración 82: Diseño adaptado de la página de Burbuja en móvil	75
Ilustración 83: Página de Conversación con diseño para tableta	76
Ilustración 84: Página de Conversación con diseño mobile-friendly	76
Ilustración 85: Interfaz responsiva de la Vista Previa	77
Ilustración 86: Interfaz responsiva de la Vista Previa para móviles	78

1. Generalidades del proyecto

1.1. Introducción

El proyecto de “Recruiting Chatbot”, consistió en el rediseño visual y funcional de una plataforma de administración para un chatbot de reclutamiento, el cual fue desarrollado con el propósito de optimizar la experiencia del usuario encargados de gestionar el proceso de selección de personal. La plataforma permite editar el estilo de chatbot para sus respectivos sitios web y, además, permitirá que los responsables monitoreen conversaciones, asimismo, analizar los candidatos y la configuración de diversos parámetros del sistema que interactúan con los postulantes.

Con la creciente incorporación de inteligencia artificial en los sistemas de reclutamiento, se consideró modernizar los procesos existentes, con el fin de mejorar la usabilidad, escalabilidad y eficiencia de la herramienta. Para ello, se llevó a cabo un análisis exhaustivo de la interfaz actual y de las historias de usuario, lo cual permitió identificar áreas de mejora y definir estrategias de rediseño centradas en la experiencia de usuario (UX).

Durante el desarrollo del proyecto se aplicaron tecnologías web para implementar estilos visuales responsivos y accesibles, así como principios de diseño centrado en el usuario. Asimismo, se colaboró con el equipo de back-end para fortalecer la seguridad en el acceso a la plataforma, con el objetivo de garantizar una navegación segura para los del usuario.

Este proyecto presenta una gran oportunidad para aplicar conocimientos adquiridos en el Tecnológico Nacional de México Campus Apizaco, al mismo tiempo que permitió desarrollar nuevas habilidades técnicas relacionadas con el diseño de interfaces, el desarrollo de back-end y la incorporación de soluciones basadas en entornos reales de trabajo.

1.2. Descripción de la empresa

GRUPO IMPULSOR DE INNOVACIÓN EN TECNOLOGÍA Y AUTOMATIZACIÓN DE PROCESOS EMPRESARIALES INNOVAHUE, S.A.S



Es una empresa de consultoría mexicana especializada en Marketing Digital, Desarrollo de Software y Capacitación en Microsoft 365. Proveemos la creatividad, el trabajo en equipo y la innovación como pilares para el crecimiento profesional y empresarial en el ámbito digital y tecnológico.

Misión

Somos una empresa dedicada a ayudar a las MiPyMEs y negocios del país a establecer y desarrollar las estrategias, el contenido digital, el software y la tecnología que necesitan para impulsar el crecimiento de su negocio.

Visión

Ser reconocida como una de las principales empresas mexicanas en ofrecer capacitación y consultoría metodológica e integral en Marketing Digital y tecnologías Microsoft, brindando soluciones completas, alineadas y acordes a las necesidades y visión de nuestros clientes

Dirección: Calle Río Guadiana 31, Colonia, Cuauhtémoc, Cuauhtémoc, Ciudad de México. C.P. 06500, la imagen de donde se encuentra su despacho la encontraran en la *ilustración 1*.

Correo: contacto@giintapeinnovahue.com

Teléfono: 56 5397 5314

Sitio Web: <https://giintapeinnovahue.com>



Ilustración 1. Despacho 204 GIINTAPE INNOVAHUE

1.3. Descripción del cargo del residente

El proyecto Recruiting Chatbot consistió en el rediseño de una plataforma de administración para un chatbot que está enfocado en procesos de reclutamiento, integrando funcionalidades de inteligencia artificial para optimizar la interacción con los postulantes.

Es decir, que el enfoque principal es mejorar la experiencia del usuario y modernizar la interfaz, el papel de la Ingeniera en Tecnologías de la Información y Comunicaciones, al cual se hace referencia como desarrolladora front-end, fue fundamental. Su función fue aplicar principios de diseño centrados en el usuario, implementar estilos visuales responsivos y garantizar la accesibilidad y consistencia visual de toda la plataforma. La desarrolladora front-end analizo la estructura de la interfaz, proponiendo mejoras visuales y funcionales, de igual manera, desarrollo soluciones que se alinean con las necesidades del sistema. Para ello, hizo uso de tecnologías web como HTML, CSS, Java Script y además de frameworks, lo que permitió que se crearan

componentes visuales intuitivos y eficientes. Además, colaboro con el equipo de back-end para asegurar una integración fluida en la lógica del sistema.

1.4. Problemática por resolver

En la etapa inicial del proyecto Recruiting, la plataforma de administración presentaba diversas limitaciones a nivel visual, que afectaban negativamente la experiencia del usuario, es decir, que estas deficiencias obstaculizaban la eficiencia operativa y la adopción del sistema por parte de los responsables del área de talento.

1.4.1. Principales problemáticas

1. Interfaz poco intuitiva y visualmente desactualizada.
2. Falta de responsividad
3. Estilos y componentes no reutilizables
4. Falta de accesibilidad y estándares de usabilidad

1.5. Objetivos

1.5.1. Objetivo General

El objetivo principal de Recruiting Chatbot es reestructurar el diseño para mejorar su funcionalidad, usabilidad y adaptabilidad a diferentes plataformas, garantizando una experiencia de usuario más intuitiva y eficiente. Este rediseño incluirá mejoras en la interfaz visual, la responsividad para dispositivos móviles y la compatibilidad con diversas páginas web, facilitando su implementación y comercialización. Además, se añadirá la opción de personalización para que las empresas puedan ajustar los colores del chatbot según su identidad corporativa, ofreciendo una mayor flexibilidad y adaptación a sus necesidades.

1.5.2. Objetivos Específicos

1. Analizar la interfaz de la plataforma Recruiting Chatbot para identificar áreas de mejora en términos de usabilidad, diseño visual y experiencia del usuario.
2. Rediseñar la interfaz gráfica que siga principios de diseño centrado en (UX/UI).
3. Implementar un diseño responsivo que garantice la correcta visualización y funcionalidad del sistema en distintos dispositivos.
4. Integrar opciones de personalización visual, permitiendo a las empresas modificar los colores y estilos del chatbot según su identidad corporativa.

1.6. Justificación

El Recruiting Chatbot busca optimizar el proceso de selección, facilitando la búsqueda y filtrado de puestos de trabajo. Su rediseño mejorará la experiencia de usuario, haciéndolo más intuitivo y accesible en dispositivos móviles.

Además, la incorporación de opciones de personalización permitirá a las empresas adaptar la apariencia del chatbot a su identidad visual, aumentando su atractivo y usabilidad. Con estas mejoras, el chatbot podrá comercializarse como una herramienta adaptable a diversas plataformas, ofreciendo una solución eficiente y flexible para el reclutamiento en línea.

2. Marco Teórico

2.1. Diseño Centrado en el Usuario

El diseño centrado en el usuario como lo menciona Domingo & Pera (2010) es una aproximación al diseño de productos y aplicaciones que sitúa al usuario en el centro de todo el proceso. Así, podemos entender el DCU como una filosofía cuya premisa es que, para garantizar el éxito de un producto, hay que tener en cuenta al usuario en todas las fases del diseño. El término diseño centrado en el usuario nació en el laboratorio de investigación de Donald A. Norman en la Universidad de California San Diego (UCSD). Su amplia difusión se produjo después de la publicación del libro *User centered system design: new perspectives on human-computer interaction*, y se consolidó posteriormente en el libro de Norman *The design of everyday things*, aunque en su edición original fue titulado *The psychology of everyday things*. Esta obra es un referente en el ámbito del DCU y las disciplinas afines como la interacción persona-ordenador, el diseño y la experiencia de usuario.

Domingo & Pera (2010) El diseño centrado en el usuario se basa en un modelo de proceso que se divide en fases o etapas. Estas etapas se realizan o llevan a cabo de manera iterativa

La investigación y análisis de los usuarios permite recoger los requisitos de usuario y, por ello, es una etapa clave en cualquier proceso de DCU. Si no se conocen los usuarios de una aplicación o producto, sus necesidades, limitaciones y deseos, es prácticamente imposible dar una respuesta adecuada a dichas necesidades y deseos teniendo en cuenta sus limitaciones y características. Este enfoque se refleja en el ciclo de la ilustración 2 que permite refinar continuamente el producto en función de la experiencia real y expectativas.

- **Diseño/Prototipado:** Se toman decisiones de diseño partiendo de su dimensión más general (arquitectura de información y diseño de interacción) hasta su dimensión más específica (diseño gráfico en detalle y micro-interacciones). Estas decisiones se documentan y se prototipan con objetivos de evaluación.
- **Evaluación:** Aquellas decisiones de diseño y procesos críticos del producto se ponen a prueba mediante métodos de evaluación que pueden involucrar a usuarios.
- **Implementación:** Una vez el diseño ha alcanzado el nivel requerido de calidad, se procede a su implementación o puesta en producción.
- **Monitorización:** Una vez lanzado el producto se estudia el uso que de él hacen los usuarios, con el fin de identificar oportunidades de mejora.

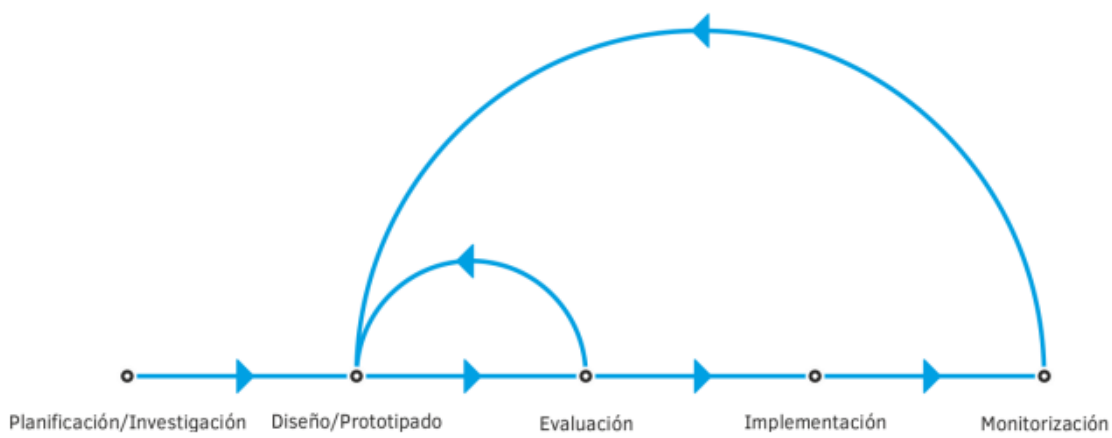


Ilustración 2. Diagrama del proceso de diseño centrado en el usuario

Según Galeano (2008) nos permite conocer su objetivo que es conocer las particularidades de los usuarios para reflejarlas en la interacción y apariencia que ofrece la interfaz y de esta manera aumentar su efectividad.

Sus orígenes del Diseño Centrado en el Usuario (DCU) es un enfoque que busca crear productos tecnológicos adaptados a las necesidades, capacidades y limitaciones reales de las personas. Sus orígenes se remontan al año 1955, y están profundamente ligados al diseño industrial, la ergonomía y la idea fundamental de que los productos deben ajustarse al ser humano y no al revés.

Durante décadas, el desarrollo tecnológico estuvo enfocado únicamente en la funcionalidad de la máquina. El objetivo principal era lograr que los sistemas funcionaran correctamente, sin fallos ni sobrecargas. Sin embargo, con el paso del tiempo, surgió la necesidad de que estos sistemas también fueran útiles, comprensibles y accesibles para quienes los utilizan. Así se pasó de un enfoque técnico a uno más humano, donde el usuario se convirtió en el centro del proceso de diseño.

Fue el diseñador industrial **Henry Dreyfuss**, quien popularizó este enfoque mediante su obra *Designing for People* (1955), en la que planteó que el diseño debía considerar las emociones, limitaciones y comportamientos de las personas. No fue sino hasta más de 30 años después que este pensamiento comenzó a ser adoptado en el ámbito de la ingeniería informática.

A partir de los años 80, con el desarrollo de la disciplina conocida como **Interacción Humano-Computadora (HCI, por sus siglas en inglés)**, el diseño de software empezó a considerar factores como la usabilidad, la legibilidad y la accesibilidad. Gracias a los avances tecnológicos en capacidad de procesamiento, memoria y representación gráfica, fue posible crear interfaces más amigables, visuales e intuitivas, lo que permitió el acceso masivo de usuarios a través de plataformas como Internet.

En este contexto, el **Diseño Centrado en el Usuario** se volvió un pilar fundamental para el desarrollo de interfaces modernas, donde ya no se espera que el ser humano se adapte a los sistemas

complejos, sino que los sistemas deben ser pensados para adaptarse de forma natural a las personas, es decir que en este contexto surge el siguiente concepto clave: la Experiencia del usuario (UX).

2.2. Experiencia del usuario (UX)

Según Galeano (2008), “La experiencia de usuario consiste en la vivencia real que tienen los usuarios con determinado producto, al relacionarse o interactuar con él. Esta vivencia incluye sensaciones y valoraciones hacia el producto, donde los diseñadores procuran que la experiencia final sea lo más agradable, positiva y satisfactoria posible, recibiendo como satisfacción final la fidelidad del usuario”.

Andrew Dillon (2001) define la Experiencia del Usuario como la suma de tres niveles: Acción, qué hace el usuario; resultado, qué obtiene el usuario; y emoción, qué siente el usuario. Aquí el autor descompone la interacción en dos niveles, acción y resultado, resaltando el aspecto emocional de la experiencia final.

Partiendo de esta comprensión funcional, es esencial que las interfaces se adapten a distintos entornos y dispositivos. Esto nos lleva al siguiente tema: el diseño web adaptable.

2.3. Diseño Web Adaptable

Se describe como una técnica de diseño y desarrollo web que, mediante el uso de estructuras e imágenes fluidas, así como de media-queries en la hoja de estilo CSS, consigue adaptar el sitio web al entorno del usuario. Toma las mejores prácticas para aplicarlas en la construcción de sitios, logrando buena calidad en las aplicaciones. La idea es que un solo sitio sea no solo adaptable a las características del recurso, sino que llegue a ser adaptativo. El diseño de la información es variable y relativo, denominado fluido. La composición se presenta en forma dinámica, puesto que cada

elemento se ajusta a las condiciones técnicas de despliegue. Para hacer un diseño web adaptativo se debe cumplir con los siguientes aspectos:

1. Diseño fluido con cuadrículas flexibles o *fluid grids*.
2. Media Queries.
3. Imágenes, objetos, videos o medios similares flexibles.
4. Fuentes tipográficas con valores relativos.

2.3.1. Beneficios del Diseño Web Adaptativo

- **Reducción de costos:** Se logra gracias a que el código es menor y no se requieren versiones distintas para su despliegue en cada uno de los dispositivos conocidos.
- **Eficiencia en la actualización:** Se utiliza una sola plantilla para la producción de la página, por lo que resulta eficiente la modificación.
- **Mejora en la usabilidad:** La legibilidad y características de uso del sitio se ajustan automáticamente en cada dispositivo.
- **Capacidad de adaptación de la interfaz:** Al ser un sitio con fluidez, la información es jerarquizada para presentar lo esencial de los contenidos para la consulta del usuario.

Alineado con la flexibilidad y mejora continua en el diseño, se hace necesario un enfoque de gestión de proyectos que favorezcan a la adaptación, la iteración y la colaboración. En este sentido, la metodología scrum se convierte en una aliada clave para el desarrollo de programas centrados en el usuario.

2.4. Metodología Scrum

La Metodología Scrum Ágil, también conocida simplemente como Scrum, combina la filosofía ágil con el modelo de administración de proyectos Scrum.

A diferencia de la idea errónea de que "ágil" significa simplemente rápido o desordenado, en realidad se refiere a que esta filosofía tiene un **enfoque incremental**, es decir que permite a los equipos desarrollar proyectos complejos en pequeños incrementos.

Por su parte, Scrum se enfoca en la planificación del trabajo dividiendo el proyecto en partes llamadas "sprints". Cada sprint tiene una duración de dos a cuatro semanas y su objetivo es construir las características más importantes y generar un producto potencialmente entregable. En iteraciones posteriores, se agregan más funciones al producto y se ajustan según los comentarios de las partes interesadas y los clientes, todo ello entre sprints.

En este proceso iterativo y organizado se representa en la ilustración 3, donde se puede observar como las actividades Scrum se integran dentro de cada sprint para asegurar la entrega continua de valor.

Por lo tanto, la Metodología Scrum es un sistema de gestión de proyectos que se basa en el desarrollo incremental en sprints y una estructura clara para la colaboración y comunicación efectiva entre los miembros del equipo y el cliente, permitiendo construir productos complejos.

2.4.1. Beneficios de la Metodología Scrum

Scrum tiene dos importantes beneficios para la optimización de los proyectos:

1. Fomenta que los productos se construyan más rápido, ya que cada conjunto de objetivos debe completarse dentro del marco de tiempo de cada sprint.

2. Como requiere una planificación frecuente y el establecimiento de objetivos de cada sprint, ayuda al equipo de trabajo a centrarse en los objetivos del sprint actual y aumentar la productividad.

Además, la Metodología Scrum traerá beneficios a nivel operativo y organizacional como:

- Flexibilidad y adaptabilidad a los cambios
- Fomentar la creatividad e innovación en los proyectos
- Costos más bajos en la producción
- Mejora de calidad de las soluciones
- Sinergia organizacional
- Satisfacción de los empleados
- Aumento de la satisfacción del cliente

2.4.2. Roles de la metodología Scrum

Para implementar Scrum de manera efectiva, es fundamental contar con un experto en la metodología, ya sea un profesional interno capacitado (Scrum Máster) o un consultor externo. Su presencia garantiza que se respeten los principios y buenas prácticas del marco Scrum, ya que una implementación incorrecta puede generar desorganización y problemas graves en el ciclo de desarrollo.

Scrum estructura los equipos de trabajo en grupos reducidos, usualmente compuestos por entre tres y nueve desarrolladores, además de dos figuras clave que pertenecen al área de gestión: el Scrum Máster y el Product Owner. Cada uno de estos roles tiene responsabilidades específicas y distintos tipos de rendición de cuentas, tanto a nivel interno como hacia el resto de la organización.

2.4.3. Scrum Máster

Es el facilitador del proceso de desarrollo de Scrum. Además de celebrar reuniones diarias con el Scrum Team, el Scrum Máster se asegura de que las reglas de Scrum se hagan cumplir y se apliquen según lo previsto. Las responsabilidades del Scrum Máster también incluyen entrenar y motivar al equipo, eliminar los impedimentos para los sprints y garantizar que el Equipo tenga las mejores condiciones posibles para cumplir sus objetivos y producir productos entregables.

2.4.4. Product Owner

Representa a las partes interesadas, que suelen ser clientes. Para garantizar que el Scrum Team siempre entregue valor a las partes interesadas y al negocio, el Product Owner (Dueño del Producto) determina las expectativas del producto, registra los cambios en el producto y administra un backlog de Scrum (una lista de tareas pendientes detallada y constantemente actualizada para el proyecto Scrum).

El Product Owner también es responsable de priorizar los objetivos de cada sprint, en función de su valor para las partes interesadas, de modo que las funciones más importantes y entregables se construyan en cada iteración.

2.4.5. Artefactos de la Metodología Scrum

Los artefactos son documentos que registran las diversas actividades y decisiones del proyecto. Existen tres artefactos:

2.4.5.1. Product Backlog

Es una lista ordenada de elementos y características que podrían ser necesarios en el producto final del proyecto. Es una fuente única de requisitos. Se actualiza a medida que se modifican o agregan

nuevos requisitos, correcciones, funciones y detalles en función de la nueva información y la lista de requisitos que obtengan de los clientes o usuarios.

El Product Owner debe clasificar los elementos de esta lista. Ten en cuenta que el hecho de que haya elementos en la lista de trabajo pendiente del producto no significa que tu equipo trabajará en ellos; más bien, los elementos de la lista del Product Backlog son opciones en las que tu equipo puede trabajar durante un sprint de Scrum.

2.4.5.2. Spring Backlog

Es un conjunto de tareas o elementos seleccionados del Product Backlog que el equipo de desarrollo se compromete a completar durante un sprint. Esto representa el trabajo planificado para alcanzar el objetivo del sprint, y este compuesto tanto por las historias de usuario seleccionadas como por las actividades necesarias para implementarlas (desarrollo, diseño, pruebas, etc.).

Por lo tanto, el backlog es creado y gestionado por el equipo de desarrollo durante la planificación del sprint (*Sprint Planning*), y puede ajustarse durante el sprint si se llegasen a descubrir nuevas tareas necesarias para alcanzar el objetivo.

2.4.5.3. Fases y Reuniones de la Metodología

El proceso de trabajo en Scrum se organiza de la siguiente manera:

1. **Organización del trabajo pendiente:** El Scrum Master identifica el trabajo que debe realizarse y lo documenta en un lugar centralizado, como una herramienta de gestión de proyectos.
2. **Planificación del sprint:** Se realiza una sesión de planificación del sprint para determinar en qué se enfocará el equipo durante ese periodo específico. Se evalúa el trabajo pendiente y se establecen los objetivos del sprint.

- # Gestión Ágil de Proyectos
-
- El diagrama ilustra el ciclo de vida de un proyecto ágil, mostrando la interacción entre el cliente, el equipo de trabajo y el equipo de desarrollo.
- Proceso Principal:**
- Trabajo pendiente (Product Backlog):** Representado por un icono de una lista.
 - Reunión de planeación del Sprint:** Representado por un icono de una reunión.
 - Trabajo del Sprint (Sprint backlog):** Representado por un icono de una pila de documentos.
 - Grupo de Trabajo (Scrum Team):** Representado por un icono de tres personas.
 - Grupo de Desarrollo:** Representado por un icono de tres personas.
 - Stand-ups diarios:** Representado por un icono de tres personas.
 - Evaluación del Sprint:** Representado por un icono de una mano dando un pulgar arriba.
 - Trabajo concluido (Product Backlog):** Representado por un icono de una lista con una marca de verificación.
- Interacción:**
- El **Cliente o Usuario (Product Owner)** interactúa con el **Grupo de Trabajo (Scrum Team)** y el **Grupo de Desarrollo**.
 - El **Grupo de Trabajo (Scrum Team)** interactúa con el **Grupo de Desarrollo**.
 - El **Grupo de Desarrollo** interactúa con el **Cliente o Usuario (Product Owner)**.
- Flujo del Proceso:**
- Trabajo pendiente (Product Backlog)
 - Reunión de planeación del Sprint
 - Trabajo del Sprint (Sprint backlog)
 - Grupo de Trabajo (Scrum Team)
 - Grupo de Desarrollo
 - Stand-ups diarios
 - Evaluación del Sprint
 - Trabajo concluido (Product Backlog)
- Logo:** INNEVO

24

2.5. Desarrollo Front-End

Como complemento técnico esencial, el desarrollo front-end permite materializar todos los principios del DCU, la UX, el diseño adaptable y la metodología ágil Scrum. ¿Qué es el Desarrollo Fron-end? Es parte del desarrollo web que involucra **todo aquello que el usuario ve**, es decir todas las partes del desarrollo que sirven para que las aplicaciones se comuniquen con el usuario.

Esto incluye toda la parte de presentación y estilizado del contenido y todo lo que respecta a la interacción con el usuario. Dentro del frontend, por tanto, está el diseño del sitio web, la maquetación y la definición de los componentes de interfaz gráfica. En general, es todo lo que contribuye a la experiencia visual del usuario, es decir que el desarrollo front- end no solo responsable de la apariencia visual, sino también de la interacción fluida entre el usuario y el sistema. Gracias a su enfoque modular y su compatibilidad con ciclos iterativos, el front-end se alinea perfectamente con la filosofía ágil y permite realizar mejoras rápidas con base en el feedback continuo de los usuarios.

2.5.1. Componentes claves del frontend

Dentro del frontend se encuentran diversos componentes o áreas de trabajo. Los más importantes son:

Estructura (HTML)

El área más básica del frontend es la creación del contenido de las páginas web. HTML (Lenguaje de Marcado de Hipertexto) es el lenguaje usado para esta definición y consiste en una estructura de marcas o etiquetas que sirven para definir elementos como párrafos, encabezados, imágenes, listas, etc.

Con HTML, lo único importante es crear una estructura adecuada para el contenido, dejando muy clara cuál es la función de cada uno de los textos, enlaces o elementos multimedia de la página web.

Estilo (CSS) para el diseño

El segundo lenguaje básico del área de frontend es el CSS (Hojas de Estilo en Cascada). Con el lenguaje CSS se define cómo se muestra el contenido estructurado mediante el HTML. Es decir, mientras que HTML sirve para expresar el contenido de la página, con CSS se define cómo debe de verse ese contenido en la pantalla del navegador.

A través de CSS, los diseñadores deben determinar colores, tipografías, espaciados, así como otros aspectos visuales del sitio web como sombras, colores de fondo, animaciones y mucho más.

Lo importante del binomio HTML y CSS es que exista una fuerte separación entre contenido y presentación, de modo que para un mismo HTML pudiéramos cambiar radicalmente el aspecto de la página solamente modificando el CSS.

Interacción (JavaScript) para la funcionalidad dinámica

Como tercer lenguaje en la web tenemos el JavaScript, que nos permite definir todos los aspectos relacionados con el comportamiento de la página y que sirve para añadir interactividad a un sitio web.

Con JavaScript se debe indicar qué pasa cuando el usuario realiza acciones sobre los elementos, hace clic en los campos de formulario, escribe en ellos, pasa el ratón sobre determinadas partes de la página, etc. Además, JavaScript nos permite manipular la página dinámicamente, de modo que se dé respuesta visual a las acciones del usuario sobre ella.

2.5.2. Herramientas y tecnologías Frontend

¿Qué son las tecnologías front-end? En el núcleo del desarrollo front-end se encuentran HTML, CSS y JavaScript, que constituyen los pilares de la web. HTML (lenguaje de marcado de hipertexto) proporciona la estructura básica de los sitios web, que se mejora y modifica mediante CSS (hojas de estilo en cascada) para el diseño. JavaScript, la fuente de elementos interactivos, permite a los desarrolladores implementar funciones complejas como animaciones, eventos de clic y manipulación de datos en tiempo real.

Además de estos conceptos básicos, las tecnologías front-end también incluyen frameworks y bibliotecas diseñadas para optimizar el desarrollo y mejorar la experiencia del usuario.

2.5.2.1. Factores para seleccionar los mejores frameworks front-end

Para poder identificar los frameworks front-end ideales para proyectos de desarrollo puede ser un desafío debido a las fortalezas y especialidades únicas de cada opción disponible.

- **Propósito y escala del proyecto:** El tipo de aplicación web o sitio web que se pretende desarrollar influye significativamente en la elección del framework. Dependiendo de si se está desarrollando una aplicación empresarial a gran escala o un proyecto más pequeño y personal, las características y capacidades que se necesitan variarán. Es fundamental seleccionar un framework que se ajuste a los requisitos específicos y la escala del proyecto.
- **Simplicidad y facilidad de adopción:** Muchos desarrolladores gestionan varios proyectos simultáneamente y se benefician del uso de marcos de trabajo sencillos e intuitivos. Los marcos de trabajo más sencillos son especialmente ventajosos para los nuevos desarrolladores, quienes podrían preferirlos a opciones más complejas. Estos marcos

facilitan un aprendizaje más fluido y una entrega de proyectos más rápida, lo cual es crucial para mantenerse competitivo y ofrecer resultados de alta calidad a los clientes.

- **Rentabilidad:** En el contexto económico actual, la rentabilidad sigue siendo un factor crucial. Los desarrolladores suelen preferir los frameworks más recientes, que no solo ofrecen funciones avanzadas, sino que también lo hacen a un precio accesible. Las soluciones rentables permiten a los desarrolladores maximizar sus ganancias y ofrecer servicios asequibles, lo que contribuye a su popularidad y adopción generalizada.

2.5.3. Tecnologías Utilizadas

Las principales tecnologías utilizadas en frontend para el proyecto fueron:

2.5.3.1. HTML

HTML sigue siendo la columna vertebral de la web, una tecnología esencial para construir la base estructural de todos los sitios web y aplicaciones web. A medida que nos acercamos a 2025, HTML continúa evolucionando, adaptándose a las necesidades del desarrollo web moderno con varias características clave:

- **Elementos semánticos:** HTML5 introdujo etiquetas HTML más descriptivas como `<article>`, `<section>`, `<nav>`, `<header>` y `<footer>`. Estos elementos hacen la web más accesible y ayudan a los motores de búsqueda y a otros agentes de usuario a comprender mejor la estructura y el contenido de los sitios web.
- **Imágenes adaptables:** con atributos como `srcset` y `sizes`, HTML permite a los desarrolladores especificar un conjunto de imágenes para diferentes tamaños y resoluciones de pantalla, lo que garantiza que los sitios web estén optimizados para todos los dispositivos;

- **API integradas:** HTML funciona perfectamente con API como la API de geolocalización, la API de arrastrar y soltar y la API de almacenamiento web, que amplían la funcionalidad de los sitios web sin la necesidad de complementos adicionales;
- **Mejoras de accesibilidad:** HTML mejora continuamente en términos de accesibilidad, con etiquetas y atributos que mejoran la navegación del sitio para usuarios con discapacidades, como los atributos 'aria-*' y las definiciones de roles.

La ilustración 4 muestra el logo de HTML, el cual simboliza su papel fundamental como lenguaje de marcado y su continua relevancia en el desarrollo web moderno.



Ilustración 4. Logo de HTML

2.5.3.2. CSS

CSS es un lenguaje de hojas de estilo que describe la presentación de un documento escrito en HTML o XML. CSS describe cómo deben representarse los elementos en pantalla, papel o en otros medios. Con el paso de los años, CSS ha evolucionado para admitir estilos complejos y diseños dinámicos. Estas son algunas de sus características principales:

- **Flexbox y cuadrícula:** Estos módulos ofrecen formas más eficientes de diseñar estructuras de diseño. Flexbox es para diseños unidimensionales, mientras que Grid es para diseños bidimensionales. Ambos ofrecen flexibilidad y control sobre la posición, el tamaño y el espaciado de los elementos.
- **Propiedades personalizadas (variables CSS):** permiten a los desarrolladores almacenar valores específicos para reutilizarlos en todo un documento, lo que facilita el mantenimiento y el cambio dinámico de las propiedades CSS.
- **Diseño adaptable:** CSS admite consultas de medios que permiten aplicar estilos solo bajo ciertas condiciones, como tamaños de pantalla, orientaciones o resoluciones específicas. Esto es crucial para crear diseños adaptables que funcionen en diversos dispositivos.
- **Transiciones y animaciones:** CSS puede manejar transiciones y animaciones, lo que permite aplicar efectos interactivos y dinámicos a los elementos de la página, mejorando la experiencia del usuario sin JavaScript adicional.

La ilustración 5 muestra el logotipo de CSS, el cual representa visualmente esta tecnología como pieza clave en la definición del estilo y diseño de la web moderna.



Ilustración 5: Logotipo CSS

2.5.3.3. Java Script

JavaScript se erige como el lenguaje de programación que impulsa los aspectos dinámicos de las interfaces web, lo que lo hace indispensable para el desarrollo web moderno. Su versatilidad y capacidad quedan demostradas por varias características clave:

- **Programación basada en eventos:** la capacidad de JavaScript para crear funciones interactivas mediante eventos como clics, movimientos del mouse y entradas del teclado lo hace fundamental para mejorar la experiencia del usuario;
- **Desarrollo full-stack:** con la llegada de Node.js, JavaScript se puede utilizar tanto en el lado del cliente como en el lado del servidor, lo que simplifica el proceso de desarrollo al utilizar un solo lenguaje en ambas plataformas;
- **Programación asincrónica:** Características como `async/await` y `Promises` permiten realizar tareas de forma asincrónica, mejorando el rendimiento de las aplicaciones web al administrar las operaciones sin bloquear el hilo de ejecución principal;
- **Bibliotecas y marcos enriquecidos:** el ecosistema de JavaScript es rico en bibliotecas y marcos, como React, Angular y Vue, que abstraen muchas de las dificultades del desarrollo web y brindan herramientas poderosas para crear aplicaciones complejas de manera más manejable.

La ilustración 6 presenta el logotipo de JavaScript, símbolo reconocido de su papel central en la creación de experiencias web interactivas, dinámicas y funcionales.



Ilustración 6: Logotipo JavaScript

2.5.3.4. Bootstrap

Los frameworks CSS son bases de código, principalmente CSS y en ocasiones también Javascript, con las que los desarrolladores pueden partir para la creación de estilos e interfaces de usuario.

Ejemplos de frameworks CSS tenemos Bootstrap o Materialize, pero hay muchos otros como Tailwind CSS, Bulma o Foundation. Estas herramientas ofrecen toda una variedad de componentes y estilos listos ya listos para usar que permiten desarrollar de manera consistente y aceleran el proceso de diseño.

La ilustración 7 muestra el logotipo de Bootstrap, uno de los frameworks CSS mas populares y ampliamente adoptados, que simboliza su relevancia como herramienta esencial en el diseño moderno de interfaces web.



Ilustración 7. Logotipo Bootstrap

3. Desarrollo

3.1.Procedimiento y descripción de actividades

3.1.1. Instalación de Visual Studio

El proceso de desarrollo del proyecto comenzó con la preparación del entorno del trabajo etapa que es fundamental para asegurarse que cada implementación de cada fase sea correcta. Se eligió Visual Studio Code (VS Code) como editor principal por su compatibilidad con tecnologías front-end y su amplio ecosistema de extensiones.

La descarga e instalación se realizó desde el sitio oficial <https://code.visualstudio.com>, mostrado en la ilustración 8, como referencia del entorno utilizado.

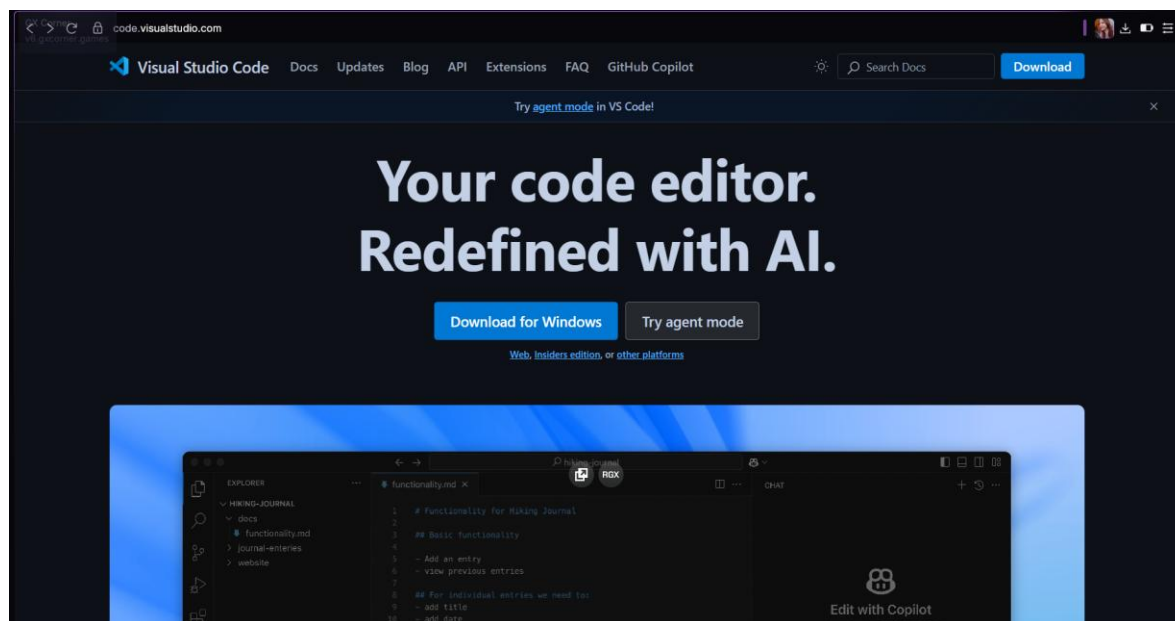


Ilustración 8. Sitio oficial para descargar Visual Studio Code

Una vez dando clic este nos abrirá el Explorador de archivos, para seleccionar en que carpeta se descargará.

Una vez hecho clic en el botón de descarga, se abrirá el Explorador de archivos para seleccionar la carpeta de destino. La ilustración 9 muestra esta acción, mostrando la ventana emergente donde se elegirá la ubicación para guardar el instalador.

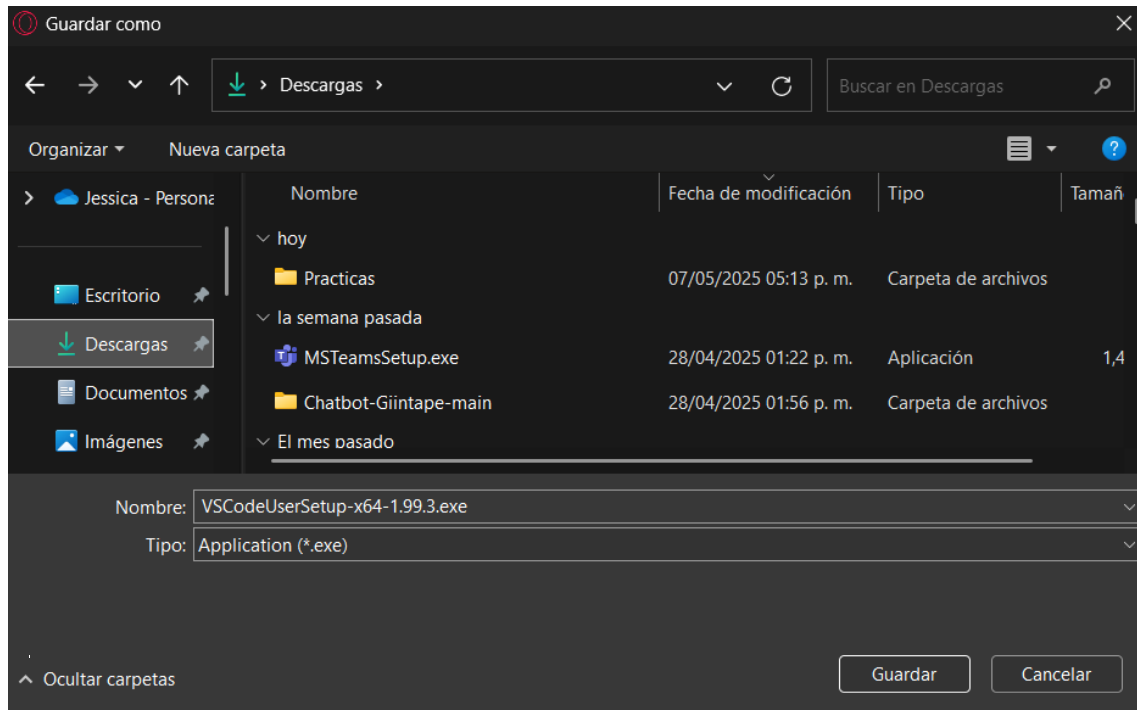


Ilustración 9: Carpeta donde se descargará

Después de completar la descarga, se procede con la instalación. La ilustración 10 mostrara la ventana inicial de asistente de instalación de Visual Studio Code, donde se comienza el proceso para configurar el entorno de desarrollo.

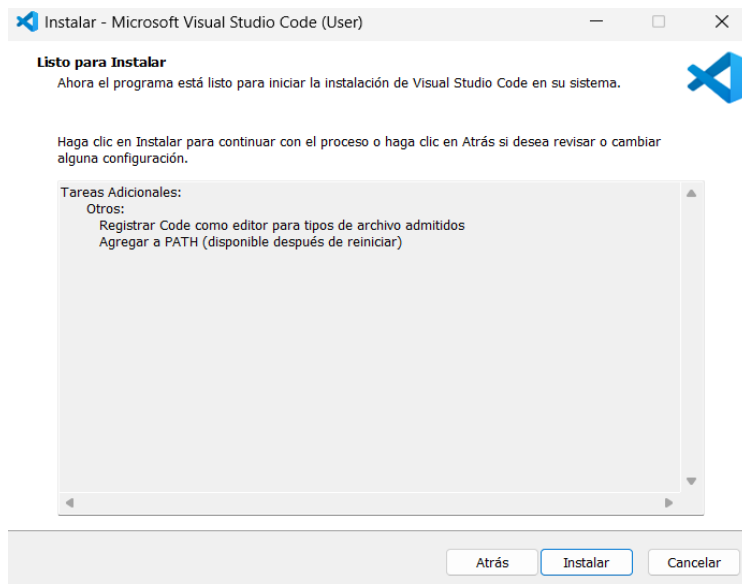


Ilustración 10: Instalación de Visual Studio Code

Finalmente, una vez completada la instalación, Visual Studio Code estará listo para su uso. La ilustración 11 presenta la pantalla de inicio de la aplicación, confirmando que el entorno de desarrollo ha sido instalado correctamente

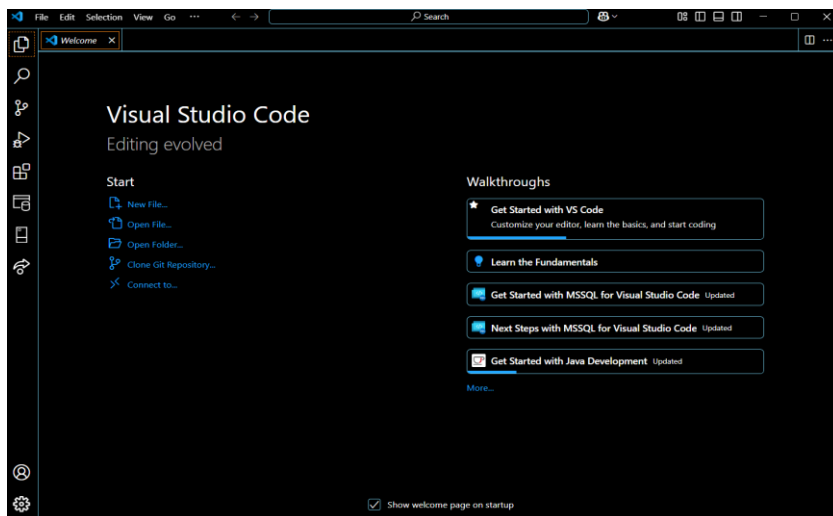


Ilustración 11: Pagina inicial del VS

3.1.1.1. Configuración de las Extensiones necesarias para el desarrollo web

Como principal tenemos el Live Para mejorar la experiencia en el área de trabajo, se procedió a instalar la extensión Live Server, una herramienta que permite previsualizar los cambios en tiempo

real directamente desde el navegador. La instalación de esta extensión se realizó desde el Marketplace integrado de VS Code, tal como se aprecia en la Ilustración 12.

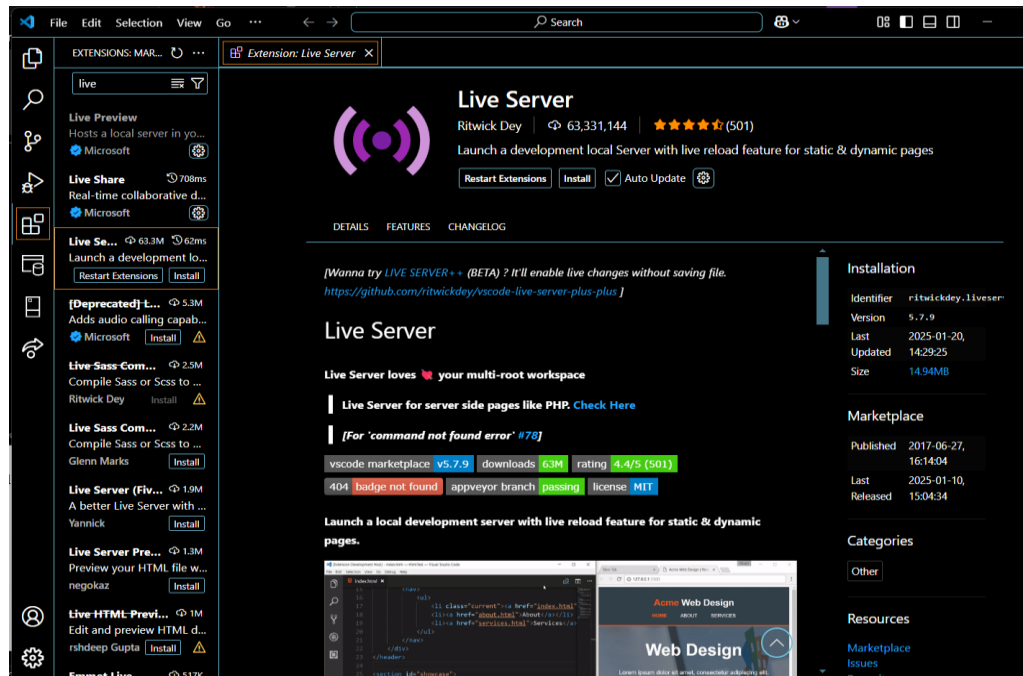


Ilustración 12: Extensión LiveServer

3.1.2. Instalación de Xampp

Posteriormente, con Visual Studio Code ya configurado, se procedió a instalar XAMPP, un paquete que facilita la creación de un entorno local para el desarrollo web, al incluir Apache, MySQL, PHP y otras herramientas.

Tal como se aprecia en la ilustración 13, se accedió al sitio oficial de XAMPP (<https://www.apachefriends.org/es/index.html>), donde se seleccionó la versión correspondiente para el sistema operativo en este caso Windows.



Ilustración 13: Sitio oficial de descarga XAMPP

Se ejecutará el instalador, la ventana de instalación nos presentará una serie de módulos, los cuales se seleccionarán los más relevantes para el proyecto.

Una vez descargado el archivo de instalación desde el sitio oficial, se ejecutará el instalador de XAMPP. En sus primeras etapas de instalación, el asistente presenta una lista de módulos como

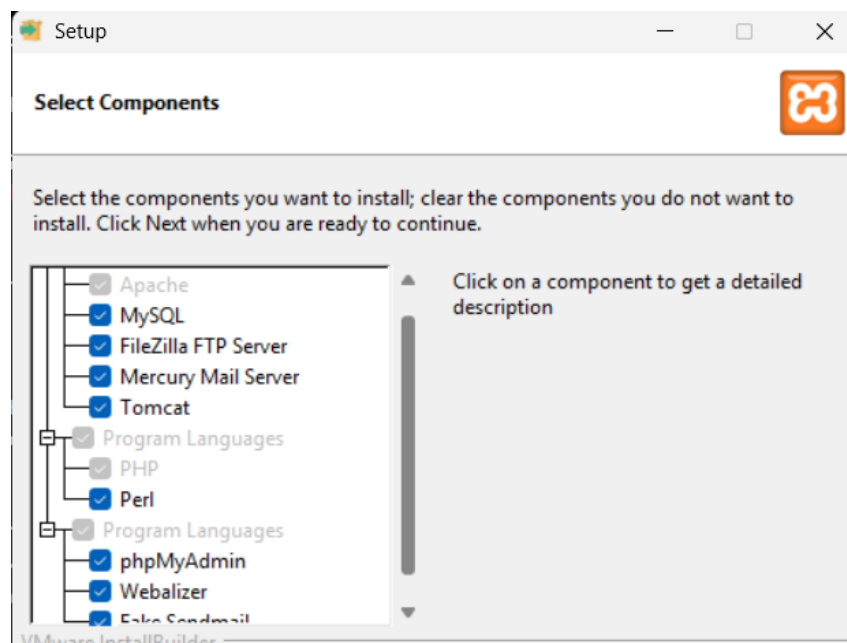


Ilustración 14: Configuración de XAMPP

se muestra en la ilustración 14 que es una ventana donde se lleva a cabo la selección. En este caso ya existe una selección por default, se dejará así y se continuará con el proceso de configuración.

Durante el proceso de instalación, como se vemos en la ilustración 15 el asistente solicita que se seleccione la ubicación del directorio donde se instalara XAMPP.

Por defecto, el sistema propone la ruta C:\\xampp, recomendada por ser una ubicación sencilla y libre de restricciones, lo que evitara que existan conflictos con otras carpetas del sistema.

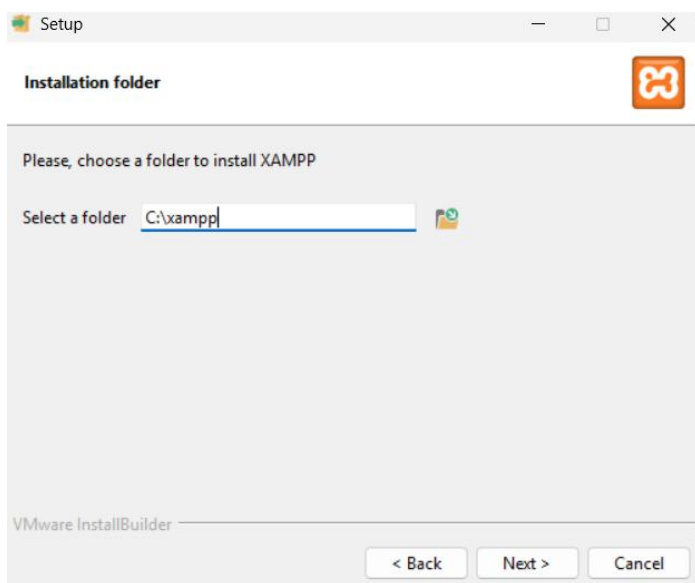


Ilustración 15. Selección de carpeta para XAMPP

Finalizado el proceso de instalación, se abrirá automáticamente el Panel de Control de XAMPP.

La ilustración 16 muestra el panel de control con los servicios con los que se iniciaran los principales servicios Apache y MySQL, necesarios para el proyecto web que se esta desarrollando.

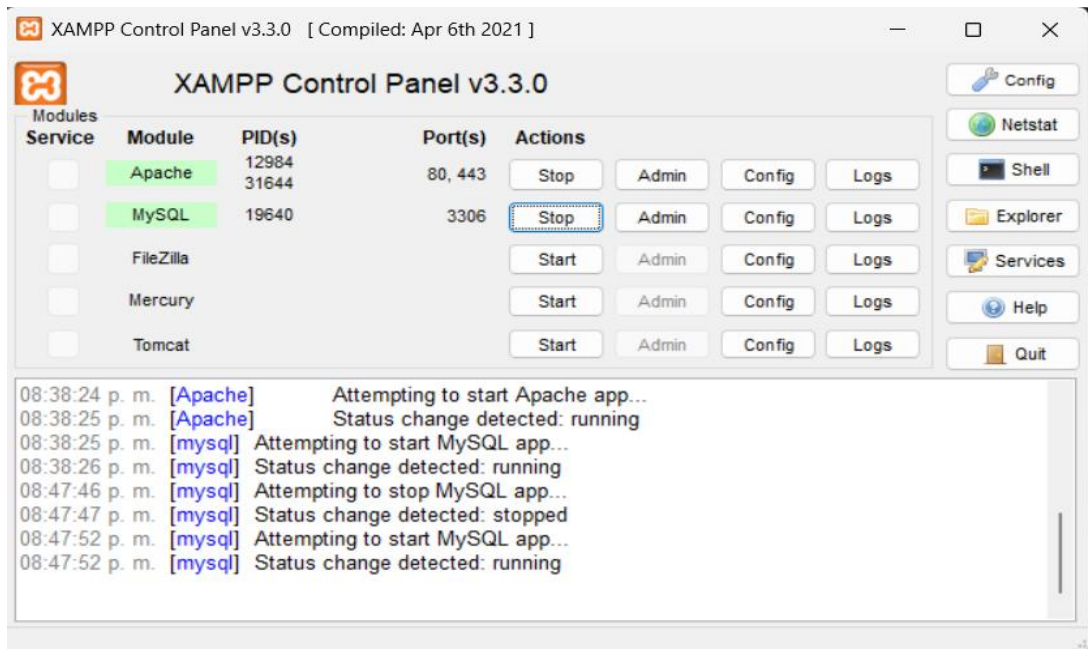


Ilustración 16: Ejecución de XAMPP

Para verificar completamente la instalación, se abrirá el navegador y se accederá a la dirección <http://localhost/>.

Si todo ha sido configurado correctamente, el navegador redirigirá automáticamente a la página de bienvenida tal como se muestra en la ilustración 17 la cual confirma que los servicios están activos, asimismo, se confirma que el entorno local esta correctamente instalado y preparado para el desarrollo del proyecto.



Ilustración 17: Prueba de localhost

3.1.3. Uso de GitHub

Una vez instaladas y configuradas las tecnologías necesarias para el desarrollo, se proporcionó una invitación al repositorio oficial del proyecto, el cual está alojado en GitHub.

Para poder interactuar con el repositorio, fue necesario iniciar sesión en GitHub tal como se muestra en la ilustración 18.

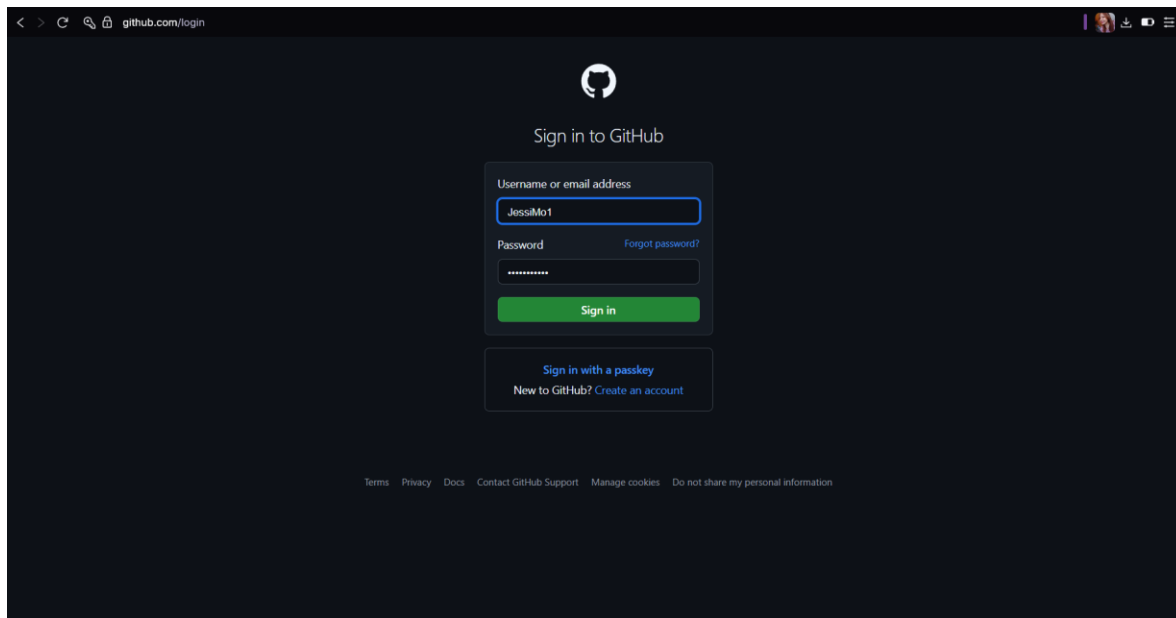


Ilustración 18: Inicio de sesión en GitHub

Después de iniciar sesión en GitHub, el siguiente paso consistió en aceptar la invitación al repositorio del proyecto, la cual fue enviada previamente por el líder del equipo tal se ve en la ilustración 19 donde se confirma la incorporación al repositorio para obtener los permisos necesarios.

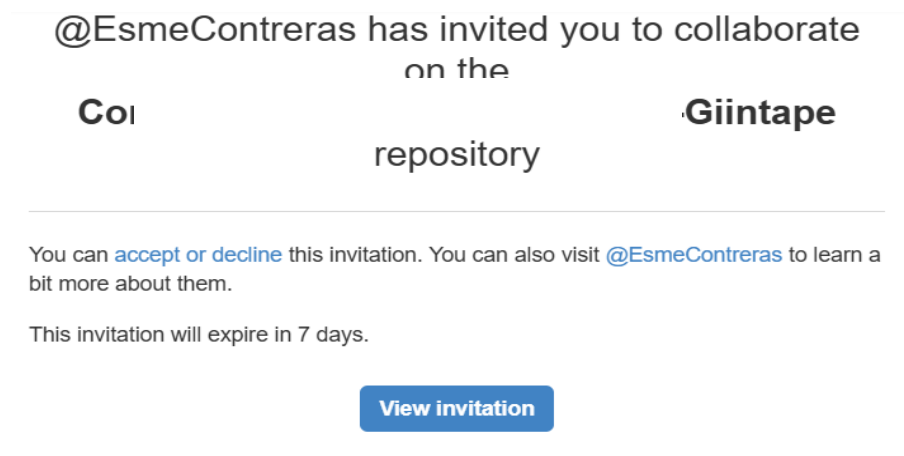


Ilustración 19: Invitación al repositorio

Una vez que se aceptó la invitación, la plataforma nos redirige automáticamente al repositorio de la empresa GIINTAPE tal cual lo muestra la ilustración 20, donde se encuentra el código.

En este repositorio, se puede acceder a todas las ramas, archivos y documentación necesaria para iniciar con el desarrollo.

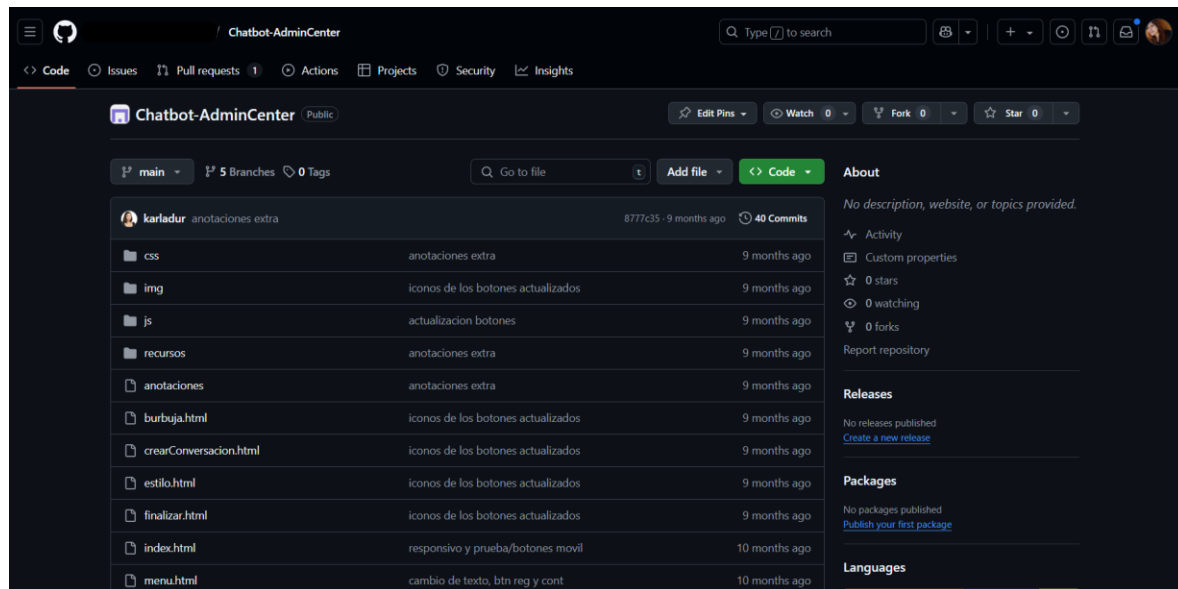
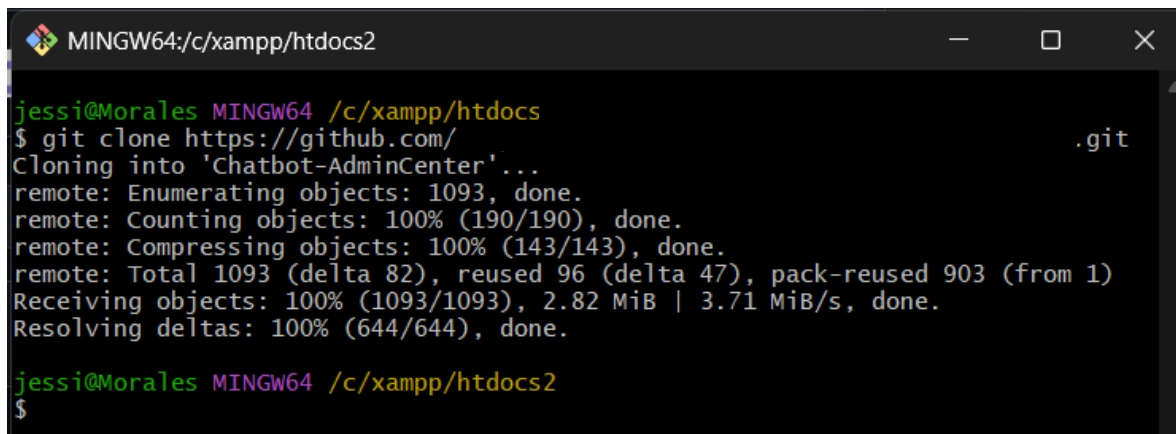


Ilustración 20: Redirección al repositorio

A continuación, se procederá con la clonación del repositorio, para esto la ilustración 21 muestra la ejecución del comando de clonación en Git Bash, que con este comando el repositorio será clonado directamente en la carpeta htdocs del entorno XAMPP, todo este proceso descargará el contenido del repositorio incluyendo ramas e historial de commits, estructura de carpetas, permitiendo comenzar con el desarrollo del proyecto.

A terminal window titled 'MINGW64:/c/xampp/htdocs2' showing the execution of a git clone command. The user 'jessi@Morales' runs 'git clone https://github.com/Chatbot-AdminCenter/...'. The output shows the cloning process: enumerating 1093 objects, counting 100%, compressing 100%, and receiving objects at 3.71 MiB/s. The terminal text is as follows:

```
jessi@Morales MINGW64 /c/xampp/htdocs
$ git clone https://github.com/Chatbot-AdminCenter/...
Cloning into 'Chatbot-AdminCenter'...
remote: Enumerating objects: 1093, done.
remote: Counting objects: 100% (190/190), done.
remote: Compressing objects: 100% (143/143), done.
remote: Total 1093 (delta 82), reused 96 (delta 47), pack-reused 903 (from 1)
Receiving objects: 100% (1093/1093), 2.82 MiB | 3.71 MiB/s, done.
Resolving deltas: 100% (644/644), done.

jessi@Morales MINGW64 /c/xampp/htdocs2
$
```

Ilustración 21: Clonación del repositorio

Una vez que se clono se nos asignó que se crearan ramas con nuestros nombres completos “JessicaMoralesAguilar” tal y como se muestra en la Ilustración 22, para poder trabajar desde nuestras ramas antes de subir a un cambio a la rama principal. Esta creación de ramas se hará directamente desde *github*.

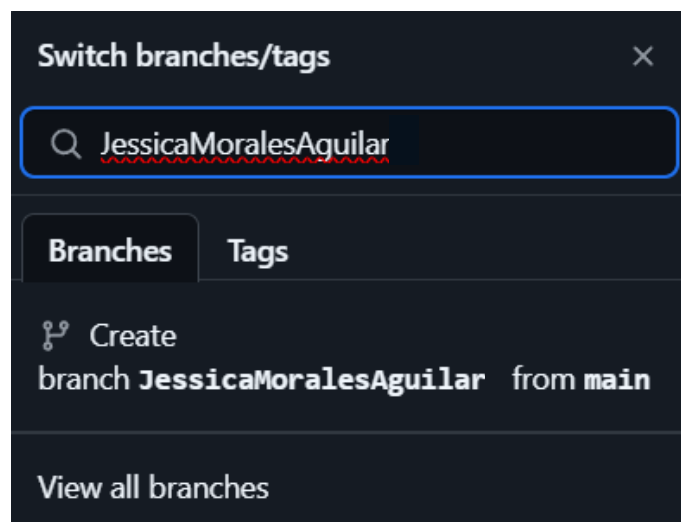


Ilustración 22. Creación de la rama

Una vez que se creó la rama, dentro del programa de Visual Studio Code se cambiara a la nueva rama que fue nombrada con nuestro nombre, para poder a comenzar con las Historias de Usuario tal y como se muestra en la Ilustración 23.



Ilustración 23: Visualización de la rama en VS

3.1.4. Sprint 0. Primera Etapa del Desarrollo

Dentro el primer sprint llamado *sprint 0* que fue un sprint para poder adaptarnos al ritmo de trabajo de la empresa, estas actividades iban relacionadas a la página del login.

3.1.4.1. Validar Correo Electrónico

En esta actividad se tiene que agregar un input como campo para la página del login donde este será el que se pueda colocar un correo electrónico tal y como se muestra en la Ilustración 24. Una vez que este el input este deberá ser configurado para que solo acepte correos electrónicos.

Captura antes de los cambios



Ilustración 24: Login sin modificaciones

Para lograr eso se utilizó el código que se muestra en la Ilustración 25, donde el type nos ayudara a especificar qué tipo de comportamiento queremos que tenga el input, en este caso fue un tipo “email”, que es el que obliga a que se espere una dirección de correo electrónico.

```
<label class="form-label">Correo</label><br />
<input
  type="email"
```

Ilustración 25: Implementación del input tipo “email”

3.1.4.2. Historia de usuario Restablecer Contraseña

Dentro de esta actividad se tuvo que añadir un enlace que se encuentre situado debajo de los inputs del login, de igual manera al dar clic en este enlace que se llamara “Restablecer contraseña” mostrara un modal para que el usuario pueda ingresar su correo electrónico eso para que el usuario reciba una notificación por correo electrónico. Eso se logró colocando una etiqueta tal y como se muestra en la Ilustración 26.

```
<!-- Enlace para restablecer la contraseña -->
<a href="#" id="link-restablecer" class="link-restablecer">¿Olvidaste la contraseña?</a>
```

Ilustración 26: Enlace de “Restablecer contraseña” agregado

Para generar el modal, que pedirá el correo electrónico para que se pueda enviar una notificación, fue trabajado tanto en código HTML mantenerlo dinámico con JavaScript como se muestra en la Ilustración 27.

Su estructura solo se base en un span para poder cerrar el modal, después para mantener una mejor estructura se colocó un div de esta manera visualmente se ve mejor para el usuario, después un texto descriptivo para mayor entendimiento, seguido de un formulario tomando en cuenta también para que el tipo de dato que se proporcione realmente sea un correo electrónico. Sin embargo, este solo es una estructura básica sin ningún funcionamiento, de igual manera se agregaron estilos para que el diseño concuerde con los colores de la empresa.

```

    <span>Restablecer tu contraseña</span>
    <span class="close" id="close-modal">&times;</span>
  </div>
  <div class="modal-body">
    <p>
      Ingresa tu correo electrónico y te enviaremos una contraseña
      aleatoria para ingresar a su cuenta.
    </p>
    <form id="reset-form">
      <input
        type="email"
        id="email"
        class="txtMail"
        placeholder="Correo electrónico"
        required />

```

Ilustración 27: Diseño del modal de restablecimiento

Para su funcionamiento dinámico, se utilizó código JavaScript que se puede observar en la Ilustración 28, de esta manera se podrá cerrar de manera dinámica, ya sea con la “X” o con “Esc”. El primer evento nos ayudará a abrir y limpiar el modal y con el export cerrar Modal, permitirá cerrar el modal.

```

/* Funcion para abrir el formulario de restablecer contraseña */
forgotPasswordLink.addEventListener("click", function (event) {
  event.preventDefault();
  /* Cada vez que se envía se limpia el input del correo */
  emailInput.value = "";
  resetPasswordForm.style.display = "flex";
});

/* Se cierra el formulario de restablecimiento de contraseña */
export function cerrarModal() {
  resetPasswordForm.style.display = "none";
}

closeFormButton.addEventListener("click", cerrarModal);

```

Ilustración 28: Código JavaScript para apertura y cierre del modal

3.1.5. Spring 1. Aplicación de Estilos Corporativos y Ajustes Visuales Iniciales

En la historia de usuario de este Spring se tenía que revisar que todas las páginas tuvieran los colores corporativos para signo de pertenencia de la empresa GIINTAPE, es decir que estos colores, fueren designados por el CEO, los colores que se utilizaron fueron naranjas y amarillos.

Como se revisó que todos las la paginas tuvieran estos colores se tuvo que revisar detalladamente la hoja de estilos, y que los colores fueran de acuerdo con la paleta de colores que nos otorgaron., como se muestra en la Ilustración 29.



Ilustración 29: Colores corporativos

3.1.6. Spring 2. Renombramiento y Personalización

Durante este sprint la primera actividad que se realizo fue el renombramiento del menú que ya se tenía, estos cambiando a Estilo, burbuja, mensaje inicial, conversación, despedida y vista previa, todo esto con el solo objetivo de que los usuarios puedan encontrar las opciones de manera más rápida al momento de querer realizar una configuración.

Para poder realizar estos cambios se tuvo que realizar la búsqueda de las etiquetas que ya estaban configuradas, es decir para realizar la búsqueda de estas etiquetas se pudieron usar la inspección de código que nos proporciona el navegador y con “Ctrl + f” para buscar estas etiquetas dentro de todas las páginas que estas contuvieran el menú. Por lo tanto, solo se hizo cambio en el texto que ya contenían las etiquetas, como se muestra en la Ilustración 30.

```

<nav class="menu-pers">
  <ul>
    <li class="estas"><a href="estilo.php">Estilo</a></li>
    <li><a href="burbuja.php">Burbuja</a></li>
    <li><a href="pantallaInicio.php">Mensaje Inicial</a></li>
    <li><a href="crearConversacion.php">Conversación</a></li>
    <li><a href="pantallaDespedida.php">Despedida</a></li>
    <li><a href="finalizar.php">Vista previa</a></li>
  </ul>

```

Ilustración 30: Cambio de nombres en el menú de navegación

Para la siguiente historia de usuario se trataba de configurar la página de estilos, puesto que esta se encarga de que los usuarios puedan elegir los colores que representen a su empresa, es decir que para lograr esto se revisó con cuidado que los que eran colores, estuvieran asociados con el tipo de dato que será color, para que así pueda salir un color picker y los usuarios puedan elegir una distinta gama de colores, como se observa en la Ilustración 31.

```

<input type="color" id="colorPrimario" value="#e39842"
oninput="actualizarColores()">

```

Ilustración 31: Campo color picker

Para esta historia de usuario se configuraría el menú de botones que se tiene, pero específicamente se cambiarían los botones que están agrupados como uno que es el de guardar/salir, se requiere que los botones queden separados, es decir, que sea un botón de guardar asimismo como un botón de salir que lo que permitirán es que el botón de guardar podrá guardar los progresos del usuario y el botón regresará al usuario la página inicial.

Como se muestra en la Ilustración 32, en la Ilustración 33 y la Ilustración 34, para realizar eso, se hicieron unas modificaciones en el código, puesto que los botones se utilizan en todas las páginas ya para darle el formato que ya tenían los anteriores botones, respetando el estilo que los líderes ya habían elegido.

```

on type= submit id= btnGuardarS class= btn
<span class="btn-text">Guardar</span>

```

Ilustración 32: Botón Guardar

```

a href= menu.php class= btnCerrar >
<span class="btn-text">Salir</span>

```

Ilustración 33: Botón salir

```

.btnCerrar,
.btnGuardarS,
.btnContinuar,
.btnRegresar {
  flex: 1;
  background-color: #fb8300;
  color: white;
  padding: 10px;
}

```

Ilustración 34: Estilo de botones

Una vez implementada la funcionalidad anterior se procedió con el desarrollo de la siguiente Historia de Usuario, esta se llevará a cabo en la página de Finalizar, donde se tendrá que definir un input para que el usuario pueda colocar una url teniendo una etiqueta de url de funcionamiento, es decir que estos cambios se realizaran a la izquierda de la página, permitiendo que aún se pueda visualizar la vista previa del Chatbot.

Para que esto se viera de la mejor manera se implementó un div para que pudiera estar separado de la vista previa, como se muestra en la Ilustración 35, para que se pudieran ajustar en los estilos se utilizó flex-direction: column que es el que nos permitirá que los elementos se coloquen en una columna marcando así la diferencia de que el input se vea del lado izquierdo y la vista previa del derecho.

```
<div class="label-func">
  <label for="input">URL de funcionamiento</label>
</div>
```

Ilustración 35: Div de la url

En el mismo enfoque de mejorar la experiencia del usuario, la siguiente historia de usuario consistió en crear un botón que fuera visible y habilitado para que el usuario pueda generar el link del código para poder implementar el chatbot.

Como se observa en la Ilustración 36, Para lograr eso, esta historia de usuario fue realizada también en la página de finalizar, este botón se encontraría debajo del input que llevaría la URL, con la etiqueta que llevara el botón llamado “generar”, se utilizó los mismos estilos para los botones que se configuraron con anterioridad.

```
<button type="submit" id="myBtn" class="btnGenerar">
  <span class="btn-text-Generar">Generar</span>
```

Ilustración 36: Botón generar

Posteriormente, se abordó la siguiente historia de usuario, centrada más en el diseño visual el diseño puesto que se eliminó un botón que era cerrar con el fin de poder minimizar la cantidad de botones sin una funcionalidad.

3.1.7. Spring 3. Optimización de Personalización y Ajustes en la Experiencia de Usuario

En esta etapa del proyecto se continuo con la implementación de nuevas funcionalidades, priorizadas según las necesidades que el usuario puede tener. A continuación, se presentarán las historias de usuario correspondientes al Sprint 3.

En una de las historias se solicita la implementación de un recuadro que permita definir la URL del logo de las empresas. Además, este recuadro debe estar acompañado de una etiqueta descriptiva que indique su propósito, facilitando la comprensión de su uso. Esta funcionalidad busca optimizar

el almacenamiento del sistema, evitando la carga innecesaria de archivos de imagen y promoviendo el uso de enlaces externos.

Como se muestra en la Ilustración 37, para lograr esto, se realizó un cambio a la página de estilos agregando un div, dentro del div de los colores para que fuera más sencillo de alinear.

```
<label for="urlLogotipo">Ingresa la URL del Logotipo</label><br>  
<div class="select-archivo">  
  <input type="text" id="urlLogotipo" placeholder="https://logo" class="input-url">
```

Ilustración 37: Código para el input

De igual manera se le agrego un botón para poder actualizar la imagen si es que se colocaba una url tal y como se muestra en la Ilustración 38.

```
<button type="button">Actualizar logo</button>
```

Ilustración 38: Código de botón actualizar

En otra historia de usuario se planeó la necesidad de reflejar una configuración que ya sería definida desde la página de estilos, es decir, que el logo que se cargue en la página de estilos también sea capaz de visualizarse en la página de burbuja, lo cual permite tener una coherencia visual para el usuario.

Para implementar esto, primero se tuvo que recuperar la url del logo que sería almacenado desde la página de estilo. Esta url se almaceno en un estado correspondiente, asegurándonos que se pueda reutilizar en la página de burbuja. Como se muestra en la Ilustración 39, de igual manera se modificó el botón de actualizar logo.

Además de que se verifico que este cambio no afectara otras funcionalidades, ni en la experiencia del usuario

```
<button type="button" onclick="previsualizarImagen()" class="update-logo-button">Actualizar logo</button>
```

Ilustración 39: Actualización del botón actualizar

Dentro del script se llaman los id con los que se nombraron, para poder almacenarlos desde el cache como se muestra en la Ilustración 40.

```
const urlInput = document.getElementById('urlLogotipo');
const chatbotIcon = document.getElementById('chatbotIcon');

if (urlInput.value.trim() !== '') {
    const logoURL = urlInput.value;

    // Guardar la URL en localStorage
    localStorage.setItem('chatbotLogo', logoURL);
}
```

Ilustración 40: Código JavaScript para almacenar

En la Ilustración 41 muestra el script que ayuda a revisar si existe un logo en el cache para poder llamarlo.

```
document.addEventListener('DOMContentLoaded', function () {
    const bubbleIcon = document.getElementById('chatBubbleIcon');

    // Revisar si hay un logo guardado en localStorage
    const savedLogo = localStorage.getItem('chatbotLogo');
});
```

Ilustración 41: Código JavaScript para recuperar imagen

Dando continuidad al desarrollo de funcionalidades relacionadas con la personalización visual del centro de administración, se abordó la necesidad de que el logo definido por el usuario fuera visualizado de manera correcta, es decir, que esta historia de usuario busca mejorar la apariencia del logo.

Para esto ya estaba creado el contenedor, sin embargo, se realizó cambios dentro del border-radius además de los width y height, por ello, también nos aseguramos de que se estuviera usando la

propiedad de object-fit: cover para que así nos aseguremos que el logotipo ocupe adecuadamente el área que ya está destinada sin que se deforme o sea recortado, como se muestra en la Ilustración 42.

```
width: 35px;  
height: 35px;  
border-radius: 40%;
```

Ilustración 42: Estilos para adecuar el logotipo

Con el objetivo de poder seguir fortaleciendo la personalización de la interfaz por parte del usuario, se volvió a elegir la implementación de herramientas que facilitan la elección visual de los estilos, es decir, que esta historia de usuario habla de la selección de una gama de colores.

Se reviso el código para comprobar que el componente tuviera un tipo que fuera color, para poder cambiar los colores desde un color picker, como se observa en la Ilustración 43.

```
<label for="colorTextoBurbuja">Color de texto</label><br>  
<div class="color-selector">  
  <input type="color" id="colorTextoBurbuja" value="#000000"
```

Ilustración 43: Color picker en burbuja

Ahora continuando con las mejoras en la interfaz, se buscó que se tuviera coherencia visual y funcional entre las diferentes páginas. En este contexto, la historia de usuario aborda la necesidad de que se elimine la burbuja del chatbot que se encuentra en la página de conversación.

Realizando esta actividad, se procedió a eliminar dicho elemento visual, puesto que no aportaba un valor funcional, permitiendo que el usuario se concentre en las configuraciones de la conversación.

Dando continuidad a las mejoras de la interfaz, también se realizaron pequeños ajustes que estarán más enfocados en hacer más clara la interacción con los usuarios, es decir, la historia de usuario indica que se hará un cambio de etiqueta.

Como se muestra en la Ilustración 44, se realiza un ajuste en la página de pantalla inicio, para poder hacer las modificaciones necesarias.

```
<label class="label-nombrechat">Mensaje inicial</label><br>
```

Ilustración 44: Cambio de nombre en la etiqueta

Continuando con los ajustes orientados a simplificar la experiencia del usuario, esta historia de usuario está redactada para poder realizar mejoras en los textos descriptivos dentro de la interfaz.

La intención de este cambio es reducir la sobrecarga visual, para poder evitar confusiones, especialmente cuando el usuario ya ha comprendido el propósito del campo a través de la interacción

3.1.8. Spring 4. Mejoras en la Configuración y Finalización

Durante el Spring 4, se dio continuidad al desarrollo del proyecto, estas nuevas historias de usuario están centrados a refinar las funcionalidades existentes y atender nuevas necesidades identificadas por el Product Owner. Las historias de usuario planteadas en este spring se orientaron principalmente a mejorar la apariencia visual de las páginas en las que se navegara para la configuración del chatbot.

Comenzaremos atendiendo aspectos relacionados con la simplicidad visual, es decir, que esta historia de usuario será una implementación de eliminación de la etiqueta del campo que tiene como referencia “Escribe un mensaje de despedida” con el objetivo de que los usuarios perciban esta acción como repetitiva.

Como se muestra en la Ilustración 45, esta historia de usuario fue realizada, buscando en la página de despedida esta etiqueta, y eliminar esa línea de código.

```
<label class="label-nombrechat">Escribe el mensaje de despedida </label>
```

Ilustración 45: Eliminación de línea de código

Continuando con las mejoras enfocadas a la experiencia del usuario se abordó la historia de usuario que hace referencia a que el campo de texto del mensaje de despedida cambie a tipo textarea, con un límite de caracteres.

Como se observa en la Ilustración 46, para poder implementar esta funcionalidad, se reemplazó el tipo input por un elemento textarea, asimismo se estableció una restricción de longitud de caracteres.

```
<textarea id="inp-depedida"
  placeholder="Gracias por usar JobHelper, es un gusto haber podido ayudarte... ¡Hasta la próxima!"
  class="input-despedida" required maxlength="280"></textarea><br>
```

Ilustración 46: Cambio a textarea y limitación

3.1.9. Spring 5. Implementación de Diseño Responsivo para Dispositivos Móviles

Dando continuidad al desarrollo del sistema, en el Sprint 5 se abordaron nuevas tareas orientadas a optimizar la experiencia del usuario desde diferentes dispositivos. En esta etapa, el enfoque principal fue la adaptación responsiva del sistema, con el fin de asegurar una correcta visualización y usabilidad tanto en tabletas como en teléfonos móviles.

A partir de pruebas realizadas en distintas resoluciones, se identificaron ajustes necesarios en las hojas de estilos, justo como ya se habrían descrito en las historias de usuario. Se aplicaron ajuste de diseño en las columnas, márgenes alineaciones, priorizando una mejor navegación.

Esta mejora que se solicita responde a la necesidad de ofrecer que la plataforma sea accesible y flexible, alineada con prácticas del front-end. Además, se buscará garantizar que las funcionalidades implementadas en los sprints anteriores se mantuvieran estables y estéticamente adaptadas en todos los formatos.

Iniciando con la primera historia de usuario, para mejorar la adaptabilidad de la aplicación esta historia está orientada específicamente en la página de inicio.

Para implementar esta mejora se tuvo que analizar las diferentes resoluciones con la que cuenta esta página, tomando en cuenta las resoluciones estándar, una vez que se revisó la página, modificaremos el archivo css, el archivo ya contaba con **media queries**, de tal modo que solo se tenía que reorganizar los elementos visuales con los que ya cuenta la página inicial.

Ahora un fragmento representativo de los ajustes, como se muestra en la Ilustración 47 y en la Ilustración 48.

```
@media (max-width: 480px) {  
  .container-chats {  
    inline-size: 79%;  
    max-width: 100%;  
    box-sizing: border-box;  
    overflow-x: hidden;  
    padding: 10px 10px;  
  }  
}
```

Ilustración 47: Ajuste del contenedor

```
@media (max-width: 480px) {  
  /*Adaptación de el contenedor y botones de Conversacion*/  
  .container-btn-edit-chat {  
    float: none;  
    margin-top: 10px;  
    width: 100%;  
    text-align: right;  
  }  
}
```

Ilustración 48: Ajuste del contenedor de los botones

Como se observa en la Ilustración 49, hubo una adaptación del texto para su reducción de tamaño en resoluciones móviles

```
@media (max-width: 480px) {
  .txtBien {
    ✓ font-size: 20px;
    ✓ top: -5px; ⓘ
  }
}
```

Ilustración 49: Adaptación del texto en móviles

Siguiendo con la línea de trabajo enfocada en la mejora de la experiencia en tabletas y dispositivos móviles, la siguiente historia de usuario es adaptar también la página de Estilo para garantizar su correcto funcionamiento en pantallas medianas y pequeñas

Este proceso implicó analizar cada elemento visual que componen la página, como los campos de color, etiquetas, botones y bloques de contenido. Además, se probó la visualización en distintas resoluciones para confirmar que los selectores de colores, los títulos y el resto de los componentes se mantuvieran accesibles y legibles, y así ofrecerle al usuario una experiencia mucho más agradable y cómoda al momento de personalizar el aspecto del chatbot desde cualquier dispositivo

Durante la implementación, se revisó nuevamente el archivo de estilos para asegurarnos de no repetir ajustes, para esto se agregaron modificaciones al código, como se observa en la Ilustración 50, en la Ilustración 51 y en la Ilustración 52.

Adaptación para la url

```
.input-url {
  flex-grow: 1;
  max-width: 200px;
  min-width: 400px;
  height: 35px;
  padding: 10px 15px;
  border: 2px solid #d9d9d9; /* Se adapta el color del borde */
  border-radius: 20px;
  background-color: #ffffff; /* Fondo blanco */
}
```

Ilustración 50: Estilo para la url

```
@media (max-width: 480px) {

    /*Estilo para input del logo */

    .input-url {
        max-width: 120%;
        min-width: 130%;
        width: 100%;
        height: 35px;
        padding: 8px 12px;
    }
}
```

Ilustración 51: Adaptación del input para dispositivos móviles

```
@media (max-width: 480px) {

    .txt-perso-chat2 {
        font-size: 20px;
    }
}
```

Ilustración 52: Modificación del Texto

Continuando con la adaptación responsiva del sistema, se trabajó en optimizar la visualización y funcionalidad de la página de Burbuja para los dispositivos móviles y Tablet. Esta sección solo está encargada de seleccionar los colores que llevarán la previsualización de la burbuja.

Durante el desarrollo se reorganizo y analizo el código con el que ya se contaba utilizando un flex-wrap y grid para asegurar la disposición de los elementos en las pantallas pequeñas, en las reglas que ya se especificaron con anterioridad. Como se observa en la Ilustración 53.

Además, se implementaron ajustes específicos para tabletas (Ilustración 54) y se disminuyó la fuente para mejorar la lectura en celulares, como se muestra en la Ilustración 55, asimismo como la ilustración 56 que muestra el cambio de tamaño del icono.

```
@media (max-width: 1300px) {
  .chat-text {
    right: 24px; ①
    padding: ► 4px 20px;
    height: 40px;
  }
}
```

Ilustración 53: Código para la resolución 1300 px

```
@media (max-width: 880px) {
  .chat-text {
    font-size: 11px;
    padding: 4px 20px;
    height: 35px;
  }
}
```

Ilustración 54: Resolución para tabletas

```
.txt-msg-chat3 {
  font-size: 14px;
}
```

Ilustración 55: Disminución de la fuente para la resolución de celular

```
.img-icone-dia {
  margin-top: 10px;
  width: 30px;
  height: 35px;
}
```

Ilustración 56: Disminución del icono representativo de la pagina

Siguiendo con el proceso de optimización responsiva, la siguiente historia de usuario nos hizo trabajar en la página de conversación, la cual es una página importante que permite la interacción con el chatbot. Esta sección requiere una revisión exhaustiva para asegurarnos si es que no existe código en las medias queries ya definidas, y así que los elementos que están en la página sean accesibles y legibles desde los dispositivos móviles y tabletas.

Como se muestra en la Ilustración 57, durante la implementación, se reorganizaron los componentes que serían las opciones de estas conversaciones para que pudieran centrarse en las pantallas pequeñas.

```
@media (max-width: 480px) {  
    
  .boton1 {  
    width: 90%;  
    font-size: 14px;  
    padding: 10px;  
    margin: 10px auto;  
    text-align: center;  
  }  
}
```

Ilustración 57: Reestructuración de los botones

Finalmente, se abordó la última historia que se abordará la adaptación responsiva de la página de vista previa, una sección donde se le permitirá al usuario visualizar como lucirá su chatbot antes de finalizar su configuración. Dado que la vista es altamente visual e interactiva, fue necesario revisar que los ajustes estuvieran correctamente para asegurar que los elementos gráficos y el texto se conservaran en su posición en los dispositivos móviles y tabletas.

Durante la implementación, se realizaron algunos cambios en el URL de funcionamiento para una mejor adaptabilidad en las resoluciones mayores, como se muestra en la Ilustración 58, y también en las resoluciones pequeñas, evidenciado en la Ilustración 59.

```

/*Estilo del link de funcionamiento*/

.link-func {
  position: relative;
  width: 95%;
  margin-left: auto;
  margin-right: -33%;
  top: 5vh;
}

.container-input {
  width: 100%;
  max-width: 100%;
}

.txtfunc {
  width: 100%;
  max-width: 70%;
  min-width: auto;
  height: 40px;
}

```

Ilustración 58: Código responsivo para tabletas

```

@media (max-width: 480px) {

  /*Estilo del link de funcionamiento*/

  .link-func {
    display: flex;
    flex-direction: column;
    align-items: stretch;
    justify-content: flex-start;
    width: 100%;
    height: auto;
    position: relative;
    padding: 0 70px;
    box-sizing: border-box;
  }

  /* Etiqueta */

  .label-func {
    text-align: left;
    width: 100%;
  }
}

```

Ilustración 59: Código responsivo para móviles

Después también se hicieron pequeñas modificaciones en los contenedores que contienen al chat para que fuera más fácil de adaptar a las pequeñas resoluciones, como se visualiza en la Ilustración 60 y en la Ilustración 61.

```
@media (max-width: 880px) {
  /*Contenedor del chatBot*/

  .chatbot-container2 {
    width: 80%;
    max-width: 350px;
    right: -28px;
    margin-left: auto;
    top: 30vh;
  }
}
```

Ilustración 60: Contenedor responsivo para tabletas

```
@media (max-width: 480px) {
  .chatbot-container2 {
    margin-right: -14%;
    position: absolute;
    top: 150%;
    width: 300px;
    height: auto;
  }
}
```

Ilustración 61: Contenedor responsivo para móviles

Como se observa en la Ilustración 62, de igual manera, para mejor visualización en el texto también se hizo una reducción en la fuente del texto.

```
@media (max-width: 480px) {
  ...
}

.txt-perso-chat {
  font-size: 22px;
}
```

Ilustración 62: Modificación de fuente

4. Resultados

4.1.Resultados sprint 0

Como resultado del desarrollo realizado a lo largo de los cinco sprints, se logró una mejora sustancial en la interfaz del chatbot. A partir de la aplicación de diseños centrados en el usuario (DCU), combinados con buenas prácticas de desarrollo front-end y la metodología ágil Scrum, se obtuvo un producto más intuitivo, accesible y adaptable.

En el sprint 0, considerado como la etapa de adaptación al ritmo de trabajo y a la estructura del proyecto. El enfoque estuvo en tareas de configuración inicial y desarrollo básico para la autenticación de usuarios.

Se añadió el campo para capturar un correo electrónico en la página de login, como se muestra en la Ilustración 63 y en la Ilustración 64 implementando validaciones desde el atributo type. Eso asegura que los usuarios ingresen direcciones correctas, forzando las buenas prácticas.

Adicionalmente se agregó la funcionalidad para el restablecimiento de contraseña, que incluyo un modal emergente para solicitar el correo electrónico del usuario. El modal fue estructurado con HTML y su comportamiento dinámico fue implementado mediante JavaScript, permitiendo cerrarlo de forma fluida, como se observa en la Ilustración 65.

Login

ID de Empresa

Correo

Contraseña

[¿Olvidaste la contraseña?](#)

[¿Aún no estás registrado?](#)

Iniciar sesión



Ilustración 63: Login actualizado

Correo

kklljjnn



Incluye un signo "@" en la dirección de correo electrónico. La dirección "kklljjnn" no incluye el signo "@".

Ilustración 64: Campo de correo electrónico

Login

ID de Empresa

Correo

Contraseña

[¿Olvidaste la contraseña?](#)

[¿Aún no estás registrado?](#)

Iniciar sesión

Restablecer tu contraseña

Ingresa tu correo electrónico y te enviaremos una contraseña aleatoria para ingresar a su cuenta.

Correo electrónico

Enviar contraseña



Ilustración 65: Demostración activa del modal

4.2.Resultados del Sprint 1

Durante este sprint se enfocó en establecer la identidad visual del sistema, asegurando que todas las páginas del sistema reflejen los colores corporativos definidos por la empresa GIINTAPE.

Se llevo a cabo una revisión minuciosa de los archivos de estilos, garantizando que los valores de color coincidieran con la paleta oficial entregada por la líder del proyecto.

Además de aplicar los estilos necesarios, se validó que la coherencia visual estuviera presente en todas las vistas principales del sistema, lo que contribuye directamente a la experiencia del usuario y al reconocimiento de la empresa.

4.3.Resultados del Sprint 2

Durante este sprint se abordaron tareas enfocadas en mejorar la experiencia de usuario a través de ajuste visuales y funcionales en distintas secciones del sistema.

Es decir que se renombro el menú principal con títulos más claros (Estilo, burbuja, Mensaje Inicial, etc.) para mejorar la comprensión de los usuarios, como se aprecia en la Ilustración 66.



Ilustración 66: Menú actualizado

Como se observa en la Ilustración 67, en la página de estilo, se habilito los Colores picker para que los usuarios seleccionen colores personalizados representativos de su marca.



Ilustración 67: Campos color picker en la página de Estilo

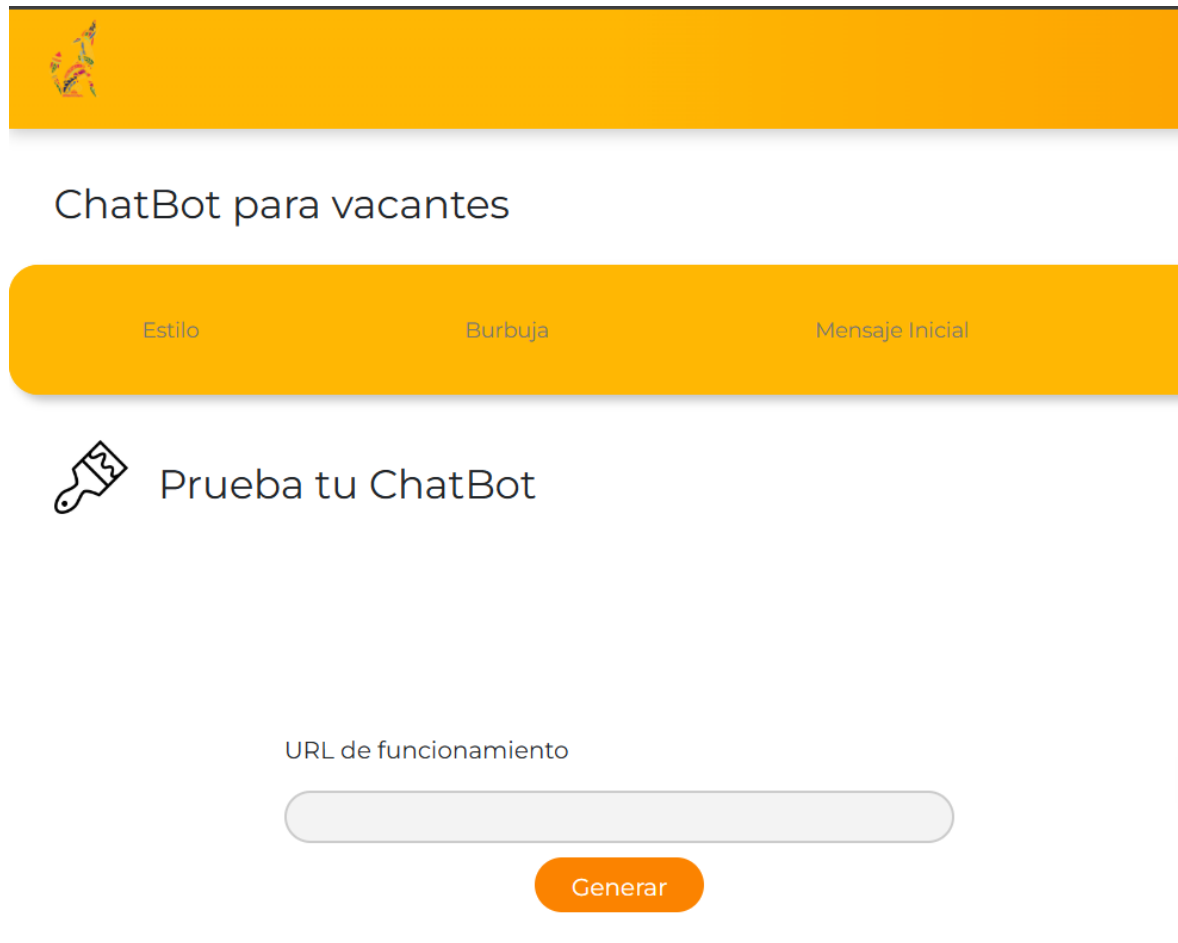
Como se muestra en la Ilustración 68, se separaron los botones de guardar y salir, brindando mayor control al usuario sobre sus acciones.



Ilustración 68: Separación de botones “Guardar” y “Salir”

En la sección de Vista previa, se añadió un campo para definir la url de funcionamiento y se ajustó el diseño utilizando `flex-direction:column` para mejorar la organización, como se observa en la Ilustración 69.

Se implementó un botón generar, que permite al usuario obtener el enlace.



The image shows a web interface for 'ChatBot para vacantes'. At the top, there is a yellow header bar with a small logo on the left. Below the header, the title 'ChatBot para vacantes' is displayed. Underneath the title is a yellow navigation bar with three tabs: 'Estilo', 'Burbuja', and 'Mensaje Inicial'. Below the navigation bar, there is a section titled 'Prueba tu ChatBot' with a paintbrush icon. This section contains a form with a label 'URL de funcionamiento' above a light gray input field. Below the input field is an orange button labeled 'Generar'.

Ilustración 69: Sección de URL agregada en la página de Finalizar

4.4.Resultados del Spring 3

Como se muestra en la Ilustración 70, en este Sprint se enfocó en el desarrollo en reforzar la personalización del sistema y mejorar la claridad visual de la interfaz.

Se añadió un recuadro con la etiqueta para que el usuario pueda definir una URL del logotipo dentro de la página de estilos junto con un botón para actualizarlo.

CHINTAPE

ChatBot para vacantes

Regresar

Estilo Burbuja Mensaje Inicial Conversación

Personaliza la interfaz de usuario del chat

Actualiza el estilo para que coincida con tu marca y tu sitio web.

Nombre visible de tu ChatBot

JobHelper

Color Primario Color Secundario Ingresa la URL del Logotipo

https://logo

Color de Acento Color de Texto

Actualizar logo

Ilustración 70: Campo para definir la URL del logo en la página de Estilo

Como se observa en la Ilustración 71, además de que se implementó la funcionalidad para que el logotipo que sea configurado en Estilo también se visualice en la página de Burbuja, garantizando coherencia visual.

De igual manera para mejor visualización se realizaron ajustes al contenedor del logotipo, sin recortes ni deformaciones. Como se observa en la Ilustración 72

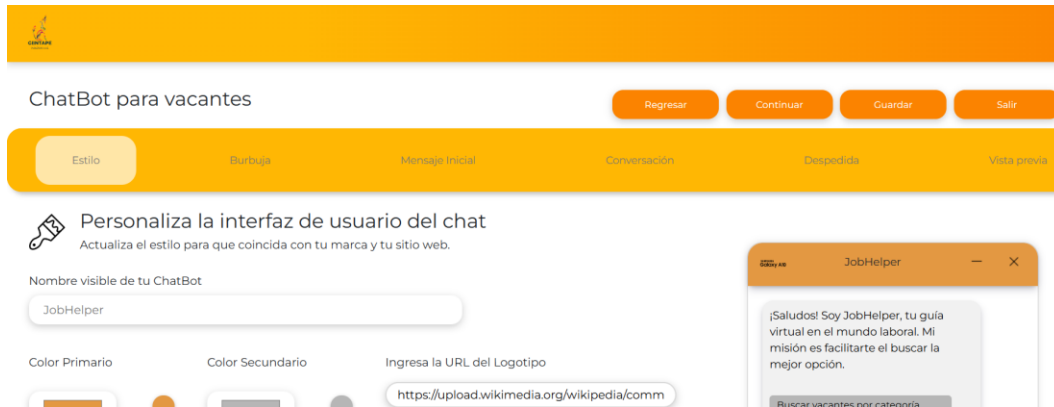


Ilustración 71: Mejoras al logotipo



Ilustración 72: Visualización del logo en página de Burbuja

Se validó que los campos de selección de color continuaran funcionando con el color picker, como se muestra en la Ilustración 73, manteniendo la posibilidad de personalizar la apariencia de la burbuja.



Ilustración 73: Color picker funcional

En la Ilustración 74 se muestra uno de los ajustes en etiquetas y campos que mejoraron la comprensión del sistema por parte del usuario final. Por ejemplo, se renombró “Mensaje saludo” por mensaje inicial.



Ilustración 74: Cambio de etiquetas en la pantalla mensaje inicial

También se eliminó el texto descriptivo innecesario “origen de la búsqueda” para reducir el ruido visual.

4.5.Resultados del Sprint 4

Durante este sprint se trabajó principalmente en mejorar la claridad de etiquetas y simplificar la interfaz para una experiencia más fluida.

Se eliminó la etiqueta de escribir un mensaje de despedida, para evitar redundancia en la información, manteniendo un diseño más limpio, tal como se muestra en la Ilustración 75.

De igual manera se actualizó el campo de mensaje despedida, cambiándolo a un textarea con límite de caracteres, permitiendo un mejor control del texto que se ingresa.



Ilustración 75: Campo de mensaje de despedida con textarea y contador de caracteres

También se eliminó el texto descriptivo de origen de búsqueda para reducir la sobre carga visual, como se puede observar en la Ilustración 76.

Origen de búsqueda



Ilustración 76: Modificación de textos descriptivos para mayor claridad

4.6. Resultado del Sprint 5

Este sprint estuvo enfocado en hacer el sistema adaptable para tabletas y dispositivos móviles, con el fin de mejorar la accesibilidad y experiencia del usuario en distintos dispositivos.

Como se aprecia en las ilustraciones 77 y 78, inicialmente se adaptó la página de inicio a resoluciones móviles, ajustando los elementos visuales y márgenes para mejorar su navegación.

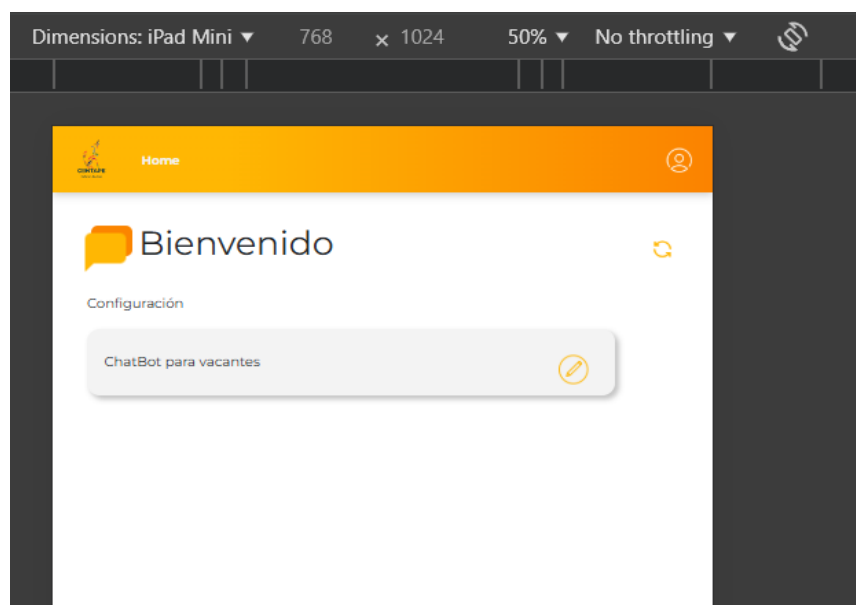


Ilustración 77: Vista responsiva de la página de Inicio en celular

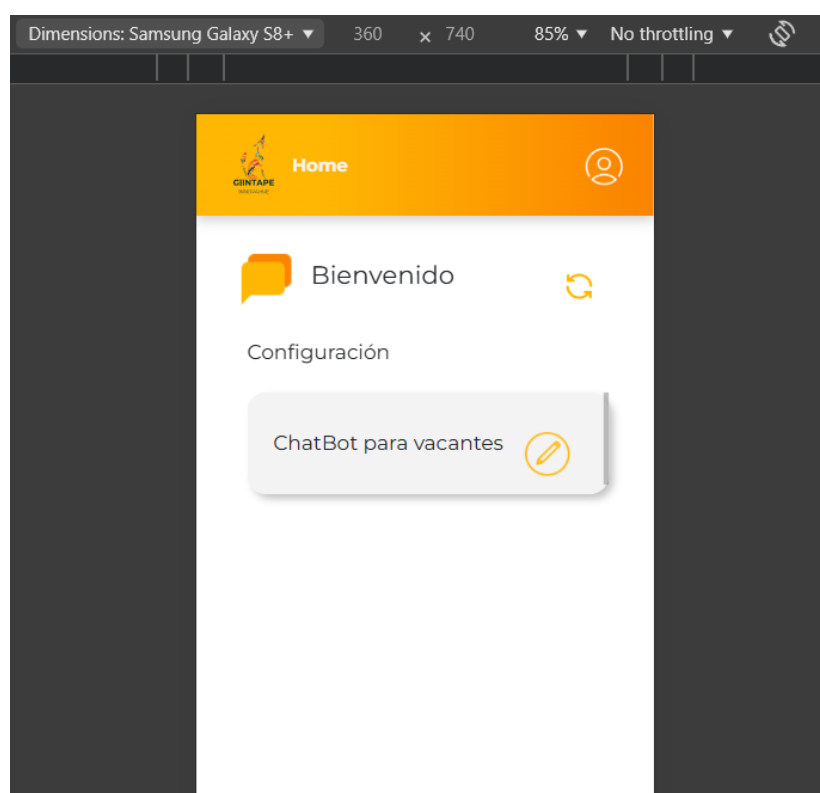


Ilustración 78: Vista responsiva de la página de Inicio en tabletas

También se trabajó en la página de Estilo, asegurando que los elementos como el color picker, inputs y etiquetas se visualizaran correctamente en resoluciones pequeñas, Tal como se muestra en las ilustraciones 79 y 80.

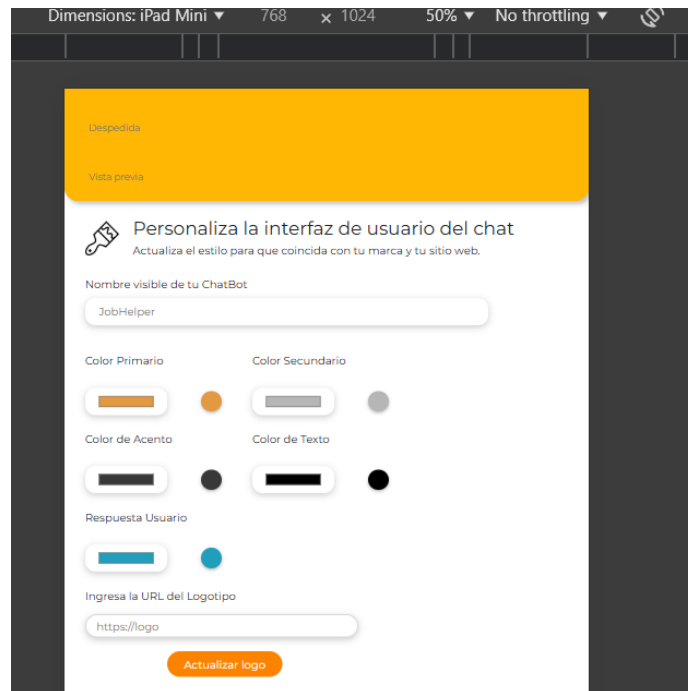


Ilustración 79: Vista responsiva de la página de Estilo en dispositivos de tabletas

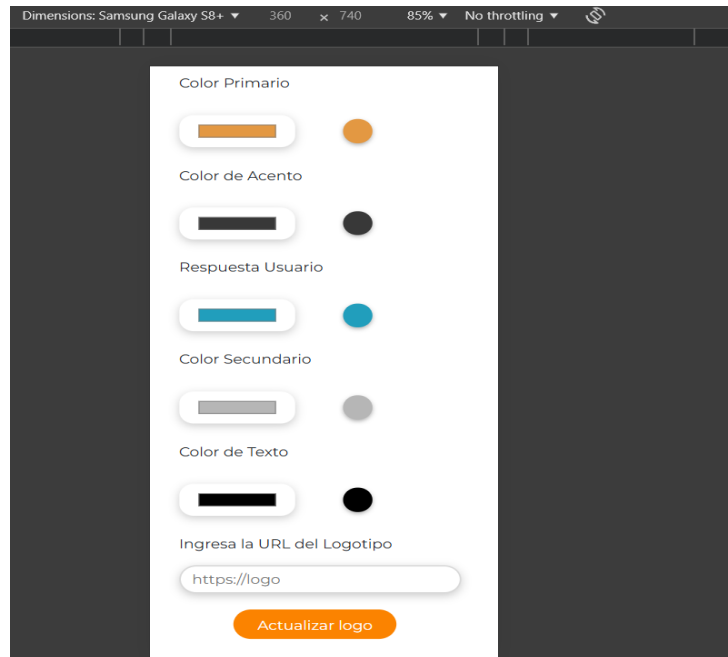


Ilustración 80: Vista responsiva de la página de Estilo en dispositivos móviles

De igual manera se realizaron ajustes en la página de burbuja, permitiendo que la configuración visual del chatbot sea intuitiva desde cualquier dispositivo, Esto se puede apreciar en las ilustraciones 81 y 82.

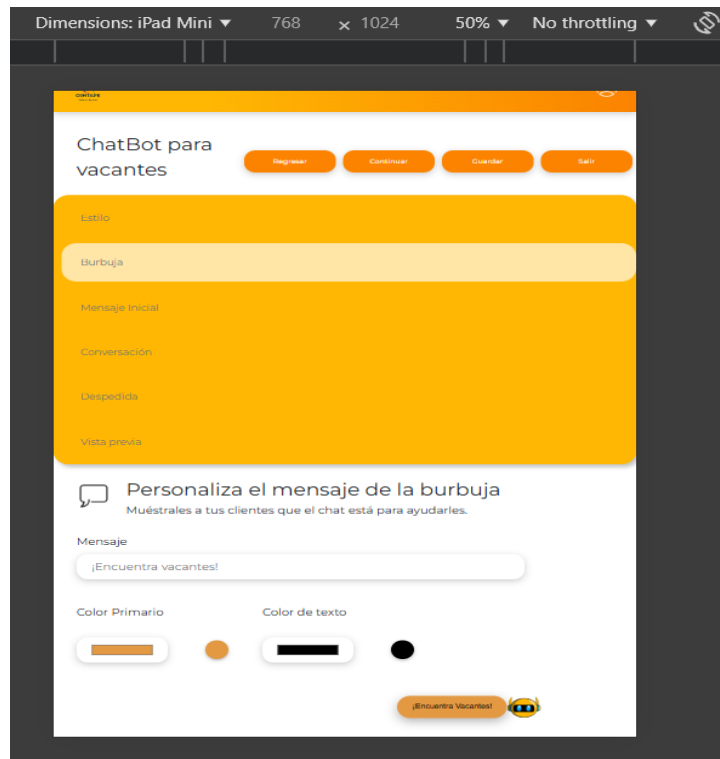


Ilustración 81: Diseño adaptado de la página de Burbuja en tableta

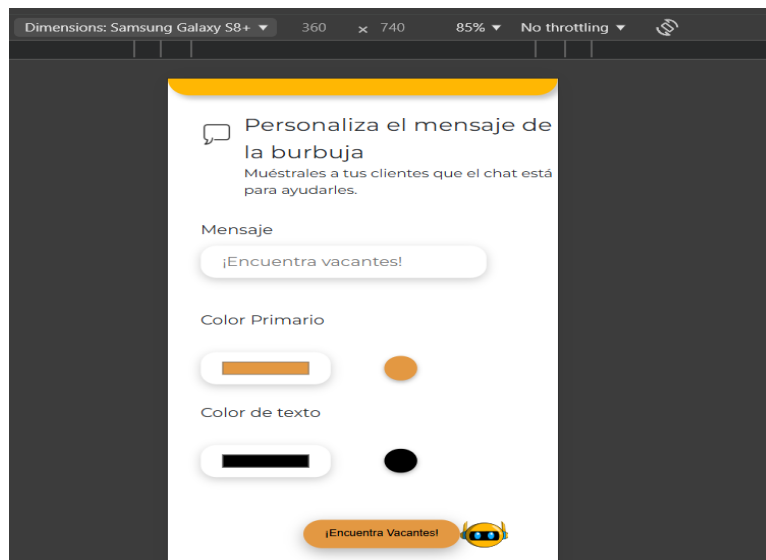


Ilustración 82: Diseño adaptado de la página de Burbuja en móvil

Se continuo con la página de conversación asegurando que las opciones de configuración estén correctamente alineadas y sean fácilmente utilizables en móviles, Como se observa en las ilustraciones 83 y 84.

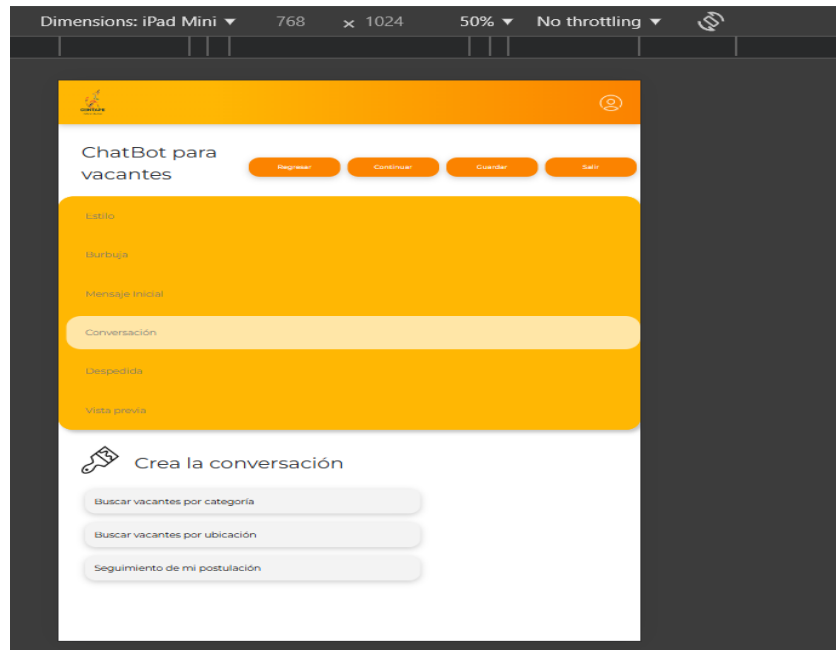


Ilustración 83: Página de Conversación con diseño para tableta

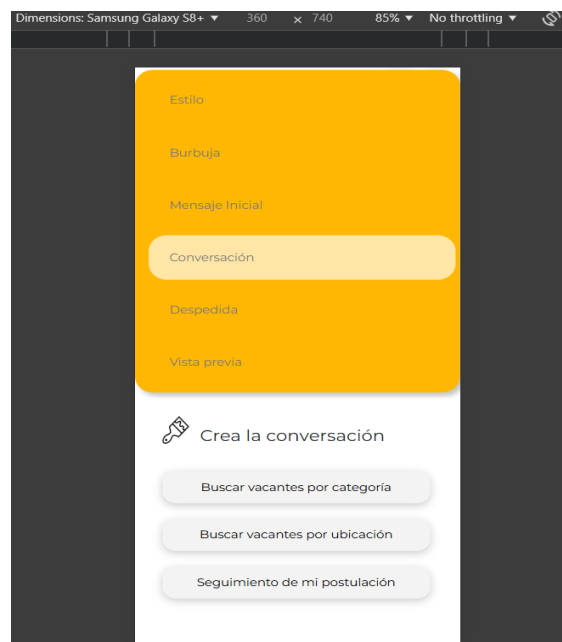


Ilustración 84: Página de Conversación con diseño mobile-friendly

Finalmente, se adaptó la página de vista previa para mantener la funcionalidad del chatbot, y su representación visual sin perder fidelidad en pantallas reducidas, tal y como se observa en la Ilustración 85 y 86.

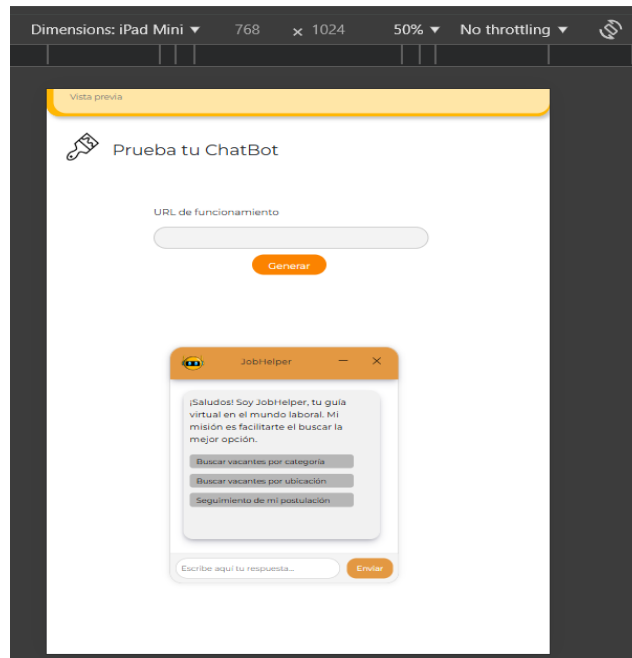


Ilustración 85: Interfaz responsiva de la Vista Previa

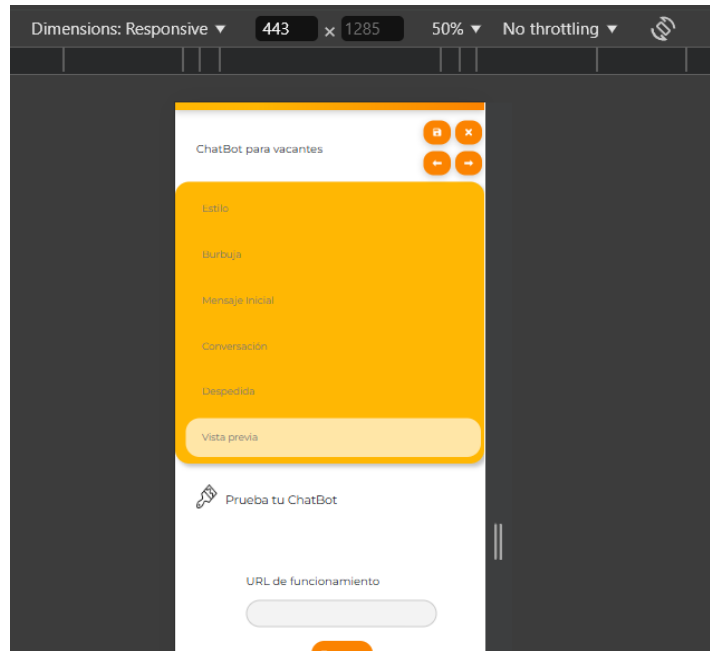


Ilustración 86: Interfaz responsiva de la Vista Previa para móviles

Finalmente, después de cada sprint, se realizó una junta con los superiores, mostrando estos cambios, es decir, exponiendo las mejoras realizadas en el proyecto y si los cambios de cada sprint eran aprobados, quedaba una persona a cargo para realizar un pull request a la rama principal y así se tuvieran estos cambios.

5. Conclusión

El desarrollo del proyecto represento una oportunidad significativa para poder aplicar y fortalecer habilidades técnicas, de diseño y metodologías ágiles en un entorno real de trabajo.

A lo largo de los distintos sprints se logró no solo cumplir con las historias de usuario planteadas, sino mejorar notablemente la usabilidad, accesibilidad y estética de la plataforma de administración.

Desde los ajustes iniciales en la pantalla de login hasta la adaptación completa del sistema para dispositivos de pequeñas resoluciones, cada iteración permitió refinar aspectos claves para la experiencia del usuario, como la personalización visual, la claridad de los textos, la organización de campos.

además del trabajo colaborativo con el jefe en conjunto con la líder de proyecto y el seguimiento de las buenas prácticas como el uso de las ramas, pull request y revisiones constantes, permitieron mantener un flujo ordenado y eficiente. La inclusión de revisiones al finalizar cada sprint facilito una validación temprana y continua, lo cual aseguro que los cambios realmente respondieran a las necesidades del producto.

En resumen, el proyecto no solo cumplió con los objetivos técnicos y funcionales propuestos, sino que también sentó una base para futuras mejoras, manteniendo como prioridad la experiencia del usuario final.

5.1.Experiencia profesional adquirida

Durante el desarrollo del proyecto obtuve una valiosa experiencia profesional como futura Ingeniera en Tecnologías de la Información y Comunicación que abarco tanto aspectos técnicos como habilidades blandas clave para el entorno laboral. La integración en un equipo de trabajo

real me permitió conocer y aplicar buenas prácticas de desarrollo ágil mediante el uso de sprints, historias de usuario y reuniones de seguimiento con los superiores.

Entre ellas se destacan:

- Dominio en diseño responsivo, adaptando interfaces a diferentes dispositivos para mejorar la accesibilidad del sistema.
- Mejoras en la experiencia de usuario (UX), mediante la reorganización visual y simplificación de textos.
- Uso de herramientas de control de versiones (Git), incluyendo la creación de ramas, migraciones, realización de pull requests y resolución de conflictos.
- Trabajo en equipo y comunicación efectiva, presentando avances, recibiendo retroalimentación y colaborando con la líder del proyecto.

6. Competencias Desarrolladas y Aplicadas

Durante el desarrollo del proyecto, se lograron fortalecer y aplicar diversas competencias técnicas adquiridas a lo largo de la carrera de Tecnologías de la Información y comunicaciones (TIC) que fueron claves para cumplir con los objetivos establecidos en cada sprint.

6.1. Competencias Específicas

Programación web: Se desarrollaron e implementaron páginas web adaptable a diferentes dispositivos, aplicando HTML, CSS, y Javascript para lograr una experiencia de usuario consistente. Se utilizó el DOM para crear componentes interactivos como botones, formularios, modales, colores, asegurando una interfaz funcional y dinámica

Interacción Humano-Computadora (IHC): Se diseñaron interfaces centradas en el usuario, priorizando la usabilidad, accesibilidad y coherencia visual. Se aplicaron principios de diseño para

lograr una navegación intuitiva, eliminando elementos innecesarios y facilitando la comprensión de las configuraciones.

Taller de ingeniería de Software. Se trabajó bajo la metodología SCRUM, organizando el desarrollo en sprints con historias de usuario bien definidas. Se llevaron a cabo reuniones de revisión con los jefes inmediatos al final de cada sprint.

6.2. Competencias genéricas

- Trabajo en equipo y colaboración
- Gestión del tiempo y planificación ágil
- Pensamiento crítico y solución de problemas
- Comunicación efectiva
- Adaptabilidad al entorno profesional

7. Fuentes de información

Arsys. (2 de Agosto de 2024). *Frontend: ¿qué es y para qué se utiliza en desarrollo web?*

Arsys: <https://www.arsys.es/blog/frontend-que-es-y-para-que-se-utiliza-en-desarrollo-web>.

Docs, M. W. (s.f.). *Uso de media queries*. MDN Web Docs:

https://developer.mozilla.org/es/docs/Web/CSS/CSS_media_queries/Using_media_queries

Galeano, R. (2008). *Diseño centrado en el usuario*. Revista q.

GitHub. (s.f.). *What is version control?* GitHub: [https://github-](https://github.com.translate.goog/resources/articles/software-development/what-is-version-control?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=tc)

[com.translate.goog/resources/articles/software-development/what-is-version-control?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=tc](https://github.com.translate.goog/resources/articles/software-development/what-is-version-control?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=tc)

Lomelí, L. (Abril de 2025). *Metodología ScRum: roles, procesos y artefactos*. Innevo:

<https://innevo.com/blog/metodologia-scrum>.

Martinez, E. L., y Ceballos, C. S. (2013). *Diseño web adaptativo o responsivo*.

Montero, Y. H. (2015). *Experiencia de usuario: principios y métodos*.

Pera, E, M., y Domingo, M. (2010). *Diseño centrado en el usuario*. Universitat Oberta de Catalunya. <https://doi.org/9-12>

Prytulenets, A. (30 de Noviembre de 2024). *16 top front end tools and technologies of 2025*.

Fively: 5ly.Co. https://5ly-co.translate.goog/blog/front-end-technologies-list/?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=tc

Translate.Goog. (s.f.). *HTML input types*. Translate.Goog: [https://www-w3schools-](https://www-w3schools-com.translate.goog/html/html_form_input_types.asp?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=tc)

[com.translate.goog/html/html_form_input_types.asp?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=tc](https://www-w3schools-com.translate.goog/html/html_form_input_types.asp?_x_tr_sl=en&_x_tr_tl=es&_x_tr_hl=es&_x_tr_pto=tc)