

BANCO DEL TIEMPO ONLINE

JHON JAIRO VALERO RESTREPO

UNIVERSIDAD TECNOLÓGICA DE PEREIRA
FACULTAD DE INGENIERIAS ELECTRICA, ELECTRONICA, FISICA Y CIENCIAS
DE LA COMPUTACIÓN
INGENIERÍA DE SISTEMAS Y COMPUTACIÓN PEREIRA
2022

BANCO DEL TIEMPO ONLINE

JHON JAIRO VALERO RESTREPO

PROYECTO DE GRADO

ING. CARLOS AUGUSTO MENESES ESCOBAR

DIRECTOR DEL PROYECTO DE GRADO

UNIVERSIDAD TECNOLÓGICA DE PEREIRA

FACULTAD DE INGENIERIAS ELECTRICA, ELECTRONICA, FISICA Y CIENCIAS

DE LA COMPUTACIÓN

INGENIERÍA DE SISTEMAS Y COMPUTACIÓN PEREIRA

2022

Agradecimientos

A mi familia y amigos que siempre me apoyaron en mi paso por la universidad y que me apoyan en mi paso por la vida.

A mis profesores y a mi asesor, quienes me brindaron herramientas para crecer personal y profesionalmente, muchas gracias por su paciencia.

A la Universidad Tecnológica de Pereira, por permitirme ser parte de ella y por brindar espacios de conocimiento accesibles a todo el mundo.

Tabla de Contenidos

- I. Introducción
- II. Generalidades
 - 1. Planteamiento del problema
 - 2. Objetivos
 - 2.1. Objetivo General
 - 2.2. Objetivos Específicos
 - 3. Justificación
- III. Marco de referencia
 - 1. Marco teórico
 - 2. Marco Conceptual
 - 3. Estado del arte
- IV Metodología
- V Planeación
 - 1. Planeación en PSP [12]
 - 1.1. Log de registro de tiempos y defectos
 - 1.2. Estándares
 - 1.3. Guión del proceso de Planeación
 - 2. Requisitos del programa
 - 2.1. Recopilación de datos
 - 2.2. Especificación de requerimientos
 - 2.3. Alcances
 - 2.3.1. Módulo de usuarios
 - 2.3.2. Módulo de intercambios
 - 2.3.3. Módulo de notificaciones
 - 2.3.4. Módulo de calificaciones
 - 2.4. Cronograma
 - 2.5. Arquitectura de Hardware y Software
 - 3. Estimación de tiempos
- VI Desarrollo
 - 7. Diseño del Software
 - 7.1. Modelo relacional
 - 7.2. Especificación del modelo
 - 7.3. Casos de uso
 - 7.3.1. Módulo de Usuarios
 - 7.3.2 Módulo de intercambios
 - 7.3.3 Módulo de Mensajes
 - 7.3.4 Módulo de calificaciones

7.4. Diagramas de Secuencia

7.4.1 Caso de uso 4, Actualizar usuario

7.4.2 Caso de uso 12, Creación de ofertas a propuestas

7.4.3 Caso de uso 24, Envío de mensajes

7.5. Casos de pruebas

7.6. Registro de defectos

8. Codificación

8.1. Estructura usada en la codificación

8.2. Tamaño de la aplicación

8.3. Imágenes del funcionamiento de la Aplicación

8.3.1. Módulo de usuarios

8.3.2 Módulo de intercambios

8.3.3 Módulo de Mensajes

8.3.3 Módulo de Calificaciones

9. Compilación y Pruebas

VII. PostMortem

1. Registro de defectos

2. Tipos de defectos

3. Resumen del Plan del proyecto y de defectos

4. Productividad de la aplicación

5. Calidad de la aplicación

5.1. Principios de calidad

5.2. Métricas de calidad

5.2.1. Yield de fase

5.2.2. Yield del proceso

5.2.3. Yield de revisión

5.2.4. Porcentaje de costo de evaluación de la calidad COQ

5.2.5. Porcentaje de fallas COQ

5.2.6. Costo de la calidad COQ

5.2.7. Relación de evaluación / fallas COQ

5.2.8. Densidad de defectos

5.2.9. Índice de calidad del proceso (PQI)

5.2.10. Tasa de eliminación de defectos de fase

5.2.11. Tasa de revisión

5.2.12. Apalancamiento de eliminación de defectos (DLR)

6. Análisis de la Calidad

7. Plan de mejora del proceso

7.1. Problemas del proceso

7.2. Sugerencias de mejora

7.3. Lecciones aprendidas

VIII Bibliografía

I. Introducción

Un banco del tiempo es un mecanismo que permite que las personas se reúnan y se ayuden entre sí, donde los participantes “depositan” su tiempo en el banco brindando ayuda y/o apoyo a otras personas. El tiempo de todos se valora por igual: una hora de servicio brindado equivale a un crédito de tiempo, sea cual sea el servicio [1][17].

Algunos ejemplos de servicios pueden ser conversaciones de idiomas, asesorías, tareas domésticas, clases de música, informática, matemática, entre otros.

Banco del tiempo es una plataforma web que permite realizar intercambios de servicios en línea, tendrá básicamente un registro de personas público para que cualquier persona pueda usarlo, a cada persona dentro de la plataforma se le registraran las horas prestadas o recibidas por sus servicios, además contará con una interfaz donde podrá ofrecer sus servicios y sumar horas y otra donde podrá solicitar algún servicio que requiera, en este caso sus horas serán restadas.

Para el presente proyecto se utilizará PSP (Personal Software Process), una metodología predictiva que hace énfasis en la documentación de todo proceso (planes, tiempos, errores, tipos de errores) y con estos datos predecir mejor el desarrollo de otros módulos, componentes o aplicaciones y mejorar la productividad personal.

La aplicación estará alojada en: <https://bancodeltiempo.net/>

II. Generalidades

1. Planteamiento del problema

De acuerdo con datos del Banco Mundial sobre los ingresos de la sociedad (Coeficiente de GINI), Colombia es uno de los países más desiguales de Sudamérica [2], desigualdad que como consecuencia afecta todos los aspectos de la vida cotidiana, una familia que no tenga cubiertas sus necesidades básicas no podrá pagar por ejemplo clases particulares para sus hijos.

Esta inequidad, que se presenta sobre todo en países en vía de desarrollo genera exclusión social debido a los bajos ingresos de las personas y que repercute en su núcleo familiar.

Por otro lado, el uso desmedido de la tecnología está aislando a las personas que la usan [3], sobre todo a los jóvenes, haciendo que se sientan más solitarios y deprimidos y que prefieran estar más tiempo en celulares y computadoras que socializando con otras personas, ser una persona sociable tiene numerosos beneficios, por naturaleza el ser humano lo es y ha sido además cooperativo y altruista [4], sin embargo faltan soluciones para propiciar una interacción más interpersonal.

Acceder a cualquier tipo de información es muy fácil ahora gracias al internet y dispositivos electrónicos, un estudiante de colegio o universidad tiene acceso a miles de artículos e investigaciones de muchas universidades o de profesionales, desafortunadamente hay también demasiada información errónea, es muy válido aprender de estas fuentes, pero es mejor aprender de la experiencia comunicada de manera más personal por otros [5], tener un mentor experto ayuda a acelerar el proceso de aprendizaje, ya que es alguien que ha encontrado solución a los mismos problemas encontrados en el colegio, universidad, trabajo o cualquier otro entorno.

Una situación bastante diferente encuentran los adultos mayores y personas pensionadas, los cuales han perdido condiciones de salud, disminución de la capacidad física y mental, además de adaptabilidad a cambios sociales y tecnológicos, lo que hace que también sean de alguna manera excluidos por la sociedad, son personas con experiencias de su larga vida y con tiempo disponible, adultos que podrían ser muy útiles en la sociedad y que debido a sus condiciones también necesitan de bastante ayuda, sobre todo relacionada con compañía y apoyo médico.

En Colombia hay algunas iniciativas de bancos del tiempo que cuentan con un espacio físico donde las personas ofrecen y solicitan servicios, estas organizaciones se encargan de hacer encuentros entre quien requiere un servicio y quien lo ofrece. En otros países se han creado páginas web [6] donde muestran la ubicación geográfica de algunos bancos del tiempo en el mundo, entre ellos Colombia. Otras han creado aplicaciones libres donde un administrador se encarga de hacer el registro de las personas y de emparejarlas, estas aplicaciones están enfocadas en bancos de tiempo con oficina física, donde además realizan entrevistas personales antes del registro oficial en el banco. También hay aplicaciones web de Estados Unidos y Europa que están más automatizadas, sin embargo la mayoría no tienen traducción para países de habla hispana. (En el estado del arte se detallan más estas aplicaciones)

¿Cómo se podría usar la tecnología para generar una colaboración mutua en las actividades, tareas, clases, etc. mencionadas anteriormente y aumentar la sociabilidad de las personas principalmente en los jóvenes?

2. Objetivos

2.1. Objetivo General

Desarrollar una aplicación web que permita a las personas registradas interactuar entre sí para ayudarse o apoyarse mutuamente permitiendo el intercambio de horas almacenadas por cada usuario.

2.2. Objetivos Específicos

- Realizar una definición de los requerimientos de software y hardware de la aplicación web.
- Realizar las etapas de análisis, diseño y diagramación de los módulos a desarrollar.
- Diseñar las pruebas para verificar el buen funcionamiento del software.
- Desarrollo de la aplicación.
- Aplicar las pruebas para medir el funcionamiento del software.

3. Justificación

Actualmente muchas organizaciones que dentro de sus actividades realizan este tipo de intercambios entre personas (bancos de tiempo) requieren de más recursos para poder operar su sistema, sería muy diferente si utilizaran las TICs para automatizar procesos, reducir personal y gastos, por ejemplo, una estructura física tiene unos gastos por el mantenimiento de la misma como los servicios públicos o arriendo, además requieren de como mínimo una persona todo el tiempo pendiente de atender a los interesados en pertenecer al banco y a los que solicitan u ofrecen sus servicios a través del banco.

Algunos estudios sobre bancos de tiempo en otros lugares y el uso que le dan sus usuarios han arrojado que en las personas que los usan hay mayor autoestima, sentido de pertenencia, apoyo social y conexiones con la comunidad. [7] Edgar S. Cahn, quien acuñó el nombre de banco del tiempo, sostiene además que los bancos de tiempo estimulan actividades que son infravaloradas por la economía dominante pero que son igual de útiles para la sociedad. [8]

Una solución en internet puede abarcar una mayor cantidad de personas, con unos gastos fijos menores y con una menor cantidad de personas dando servicio a los usuarios del banco.

Otra de las ventajas de tener un sistema online es que estará disponible las 24 horas para su uso y para ser usado por cualquier persona en el mundo con una conexión a internet y un computador o dispositivo móvil.

Debido a que muchas de las plataformas que existen actualmente tienen un enfoque en otra cultura, es relevante destacar que la aplicación se desarrollará para la situación social y económica que vive actualmente la población Colombiana (sin embargo estará abierta a otros países de habla hispana).

III. Marco de referencia

1. Marco teórico

Banco del tiempo

Un banco del tiempo es un sistema de intercambio de servicios por servicios donde se usa como medida el tiempo en vez del dinero para contabilizar las horas de los servicios prestados o recibidos [1].

Ingeniería de Software

Es la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento de software, y el estudio de estos enfoques, es decir, la aplicación de la ingeniería al software, ya que integra matemáticas, ciencias de la computación y prácticas cuyos orígenes se encuentran en la ingeniería [9].

World Wide Web (WWW)

Es básicamente un medio de comunicación de texto, gráficos y otros objetos multimedia a través de Internet, es decir, la web es un sistema de hipertexto que utiliza Internet como su mecanismo de transporte o desde otro punto de vista, una forma gráfica de explorar Internet [10].

Bases de datos

Colección de datos interrelacionados almacenados juntos en uno o más archivos computarizados [11]. Otra definición es: colección de datos organizados acorde a una estructura conceptual describiendo características de los datos y las relaciones entre sus entidades correspondientes, apoyando una o más áreas de aplicación.

2. Marco Conceptual

Proceso Personal de Software (PSP - Personal Software Process)

Metodología predictiva y procesos definidos y de mejora continua, PSP [12] hace énfasis en la documentación de todo proceso (planes, tiempos, errores, tipos de errores) y con estos datos predecir mejor el desarrollo de otros módulos, componentes o aplicaciones con el fin de construir productos de alta calidad a tiempo y con el presupuesto establecido.

A medida que se avance en el proyecto se irá explicando paso a paso el proceso de planeación, levantamiento de requerimientos, diseño, desarrollo y pruebas donde se utilizará PSP como metodología de desarrollo y cuyas prácticas sugieren el uso de estándares y métricas con el fin de tener una mejora continua en el proceso de desarrollo, Este documento incluirá también formatos de registros de tiempos de cada una de las fases del desarrollo, registros de defectos, planes de mejora del proceso y resultados de algunas técnicas estadísticas que usa PSP para mejorar la estimación, planeación y calidad.

Las siguientes definiciones son usadas por la metodología PSP [12]

- **Proceso:** Secuencia de pasos que se deben seguir para realizar una tarea determinada
- **Proceso definido:** Secuencia documentada de pasos para hacer un trabajo específico. Los procesos se definen para los trabajos repetitivos.
- **Plan:** Los planes incluyen los pasos del proceso y otros elementos para llevar a cabo el producto, como son recursos necesarios, roles de los participantes del proyecto, plan de calendario, presupuesto, metas y objetivos, compromisos del proyecto, riesgos identificados.

- **Fases:** Estados del proceso, PSP tiene 3, Planeación, Desarrollo (Diseño, Revisiones de diseño, Codificación, Revisiones de código, Compilación y Pruebas), Postmortem
- **Guiones (Scripts):** Descripciones expertas que guían la aplicación de un proceso
- **Formatos:** Proporcionan un marco adecuado para la recopilación y almacenamiento de datos del proceso
- **Métricas:** Cuantifican el proceso y el producto
- **Estándares:** Proporcionan definiciones precisas que guían el trabajo , la recopilación y uso de datos del proceso
- **Defecto:** Cualquier cosa que deba ser solucionada antes de que el producto esté completado, puede estar en los requerimientos, diseño o un problema de implementación.
- **Tiempo de interrupción:** Tiempo destinado en el trabajo que no está definido en el plan ni definido por el proceso. Algunos ejemplos pueden ser llamadas o descansos.
- **Calidad:** La calidad de un producto se mide en términos de la cantidad de defectos (ej. 5 defectos por cada 1000 líneas de código)
- **Yield:** Porcentaje de defectos en el producto que se eliminan en una fase particular o grupo de fases.
- **Yield de fase:** Porcentaje de defectos removidos durante una fase.
- **Yield del proceso:** Porcentaje de defectos removidos antes de entrar en la fase de compilación o pruebas
- **Yield de revisión:** Porcentaje de defectos en el programa encontrados durante la revisión.
- **Porcentaje de costo de evaluación de la calidad COQ:** Porcentaje de tiempo de desarrollo empleado en revisión de diseño y código
- **Porcentaje de fallas COQ:** Porcentaje de tiempo de desarrollo empleado en compilación y pruebas.
- **Costo de la calidad COQ:** Porcentaje de tiempo dedicado a realizar tareas de evaluación y corrección. Las principales métricas de COQ son:

- **Relación de evaluación / fallas (COQ):** Proporción de tiempo invertido en tareas de evaluación con respecto al tiempo invertido en tareas de corrección de fallas.
- **Densidad de defectos:** Número de defectos detectados por unidad de tamaño (ejemplo: número de defectos / Líneas de código).
- **Tasa de eliminación de defectos de fase:** Número de defectos encontrados por hora en la fase.
- **Tasa de revisión:** Tamaño del producto revisado por hora.
- **Apalancamiento de eliminación de defectos (DLR):** Medida de la eficacia relativa de la eliminación de defectos entre dos fases cualquiera del proceso.
- **Índice de calidad del proceso (PQI):** Métrica derivada para determinar la calidad de un proceso de desarrollo de software. El valor del PQI es el producto de cinco valores de componentes de perfil de calidad:
 - **Calidad de diseño:** Proporción del tiempo del diseño con respecto al tiempo de codificación.
 - **Calidad de revisión del diseño:** Proporción de tiempo de revisión del diseño con respecto al tiempo del diseño.
 - **Calidad de revisión del código:** Proporción del tiempo de revisión del código con respecto al tiempo de codificación.
 - **Calidad de codificación:** Proporción de defectos de compilación con respecto al tamaño del producto.
 - **Calidad del programa:** Proporción de defectos en pruebas con respecto al tamaño del producto.

3. Estado del arte

Actualmente existen muchos bancos de tiempo que funcionan con oficinas o sedes físicas donde las personas deben presentarse de manera presencial, algunas utilizan software de escritorio, hay otros bancos que tiene un proceso más automatizado usando software pero que se limitan a una zona geográfica específica, estos son algunos ejemplos de bancos de tiempo e investigación sobre los mismos:

Time Overflow [13]

Es una iniciativa europea cuyo enfoque está en los bancos de tiempo con oficinas físicas, este banco tiene perfiles de usuario administradores, cada administrador crea el banco en una zona geográfica específica (ciudad o barrio), las personas que usen el banco deben seleccionar el lugar donde están para hacer intercambios con estas personas.

Una limitante que tiene Time Overflow es que los usuarios solo pueden hacer intercambios con usuarios de la misma zona, en vez de poder interactuar con todos los usuarios registrados en otras ciudades o países. Tiene como ventaja la opción de crear perfiles administradores que apoyan y gestionan a otros usuarios (Funcionalidad que para el presente proyecto se considerará para una fase de desarrollo futura). Adicionalmente también funciona a través de internet y tiene una aplicación móvil.

Time Banking UK [14]

Time Banking UK es una plataforma muy similar a la de Time Overflow, su enfoque es también por zonas geográficas y también tiene un mapa donde se pueden buscar Bancos de Tiempo, funciona principalmente en el Reino Unido.

RERS (Reseaux d'Exchanges Reciproques de Savoirs) [15]

Red de intercambio de conocimientos recíprocos en Lausana y la Suiza francófona, fue creada en 1989 en Malley, la red de Lausana fue una de las primeras redes para el intercambio recíproco de conocimientos en Suiza, en 2019, la red celebró su 30 aniversario.

La propuesta de RERS es similar a la de los bancos del tiempo, solo que en RERS donde se intercambia formación (ej. una hora de formación por una hora de formación)

TimeBanks.org [16]

TimeBanks.org fue fundada en Estados Unidos en el año 1995, funciona de una manera similar a Time Overflow y a Time Banking UK, es decir, su enfoque es generar una red de Bancos de tiempo independientes.

TimeBanks.org creó un software llamado Community Weaver, el cual está en su tercera versión (año 2021) y permite hacer solicitudes/ofrecimientos de servicios, seguimiento de horas y de servicios solicitados u ofrecidos, mensajería entre usuarios, envío de invitaciones a unirse al banco entre otros. El idioma del software es el inglés, sin embargo permite traducción a través de un plugin de Google.

Para usar el software Community Weaver se debe crear una cuenta en <https://actionhub.timebanks.org/>, luego se debe llenar un formulario y aceptar un acuerdo de licencia, luego se debe hacer un pago de 79 dólares los cuales son para la instalación inicial del software, hosting y soporte técnico, después del primer año se debe seguir pagando una cuota anual.

The Time Of Our Lives [17]

El tiempo de nuestras vidas es una evaluación de 2 años realizada por la Universidad de East Anglia en el Reino Unido sobre los bancos de tiempo (Time

Banking UK principalmente, mencionado anteriormente en este documento). Evalúa básicamente los impactos y resultados de los bancos de tiempo en términos de su capacidad para superar la exclusión social.

Esta evaluación encontró que:

- Los bancos de tiempo atraen exitosamente a personas excluidas socialmente, como por ejemplo personas con bajos ingresos, personas jubiladas, personas discapacitadas, personas con enfermedades prolongadas, mujeres y grupos étnicos británicos no blancos.
- Los bancos de tiempo atraen exitosamente a personas que no se involucrarían en voluntariado tradicional, por ejemplo para el caso de personas con empleos no formales, la participación en Bancos de tiempo es casi el doble (78%) comparada con voluntariado tradicional (40%).
- La principal motivación para unirse a bancos de tiempo fue ayudar a otras personas, hacer amigos o conocer otras personas, ganar créditos de tiempo fue la razón menos motivadora para unirse a bancos de tiempo.
- El 91% de los participantes sintieron que en su banco de tiempo estaban construyendo amistades y confianza.

Esta evaluación también analizó los usos que daban otros banco del tiempo a las diferentes comunidades en el mundo y su adaptación en el contexto del Reino Unido, se encontró que:

- En Estados Unidos, Japón y China su uso tiene un enfoque hacia servicios médicos (sobretudo en relación a una población que envejece)
- Algunos Bancos de tiempo en Estados Unidos se han asociado exitosamente con tiendas de segunda mano con el fin de intercambiar tiempo acumulado de brindar servicios por bienes.
- En Chicago, Estados Unidos se ha desarrollado un modelo educativo de tutores que involucra a padres de familia y sus hijos, los padres deben

dedicar tiempo en actividades de tutoría y a cambio sus hijos reciben una computadora reacondicionada.

A Time Bank System Design on the Basis of Hyperledger Fabric Framework [18]

Artículo realizado por estudiantes de la Universidad Nacional de Taiwán, ubicada en Taipei, Taiwán, el cual plantea una arquitectura para el desarrollo de bancos de tiempo usando Cadena de bloques (BlockChain), la cadena de bloques se usaría para las transferencias de tiempo realizadas entre los usuarios que intercambien servicios en el banco, lo que haría más segura la transferencia de tiempo. (La cadena de bloques será considerada en un desarrollo futuro del proyecto de banco de tiempo Online)

IV Metodología

- Se recolectó información sobre requerimientos de usuario, realizando entrevistas a estudiantes de la Universidad Tecnológica de Pereira.
- Para la planeación, especificación de requerimientos, diseño, codificación y pruebas de la aplicación se utilizó PSP (Proceso Personal de Software), metodología definida anteriormente en el marco conceptual.

PSP no especifica como tal una metodología de diseño, así que se realizaron algunos diagramas UML para dar soporte a la metodología seleccionada.

V Planeación

1. Planeación en PSP [12]

Para la planeación se utilizó un Guión del Proceso, estos guiones son definidos ya por la metodología, cada uno de los pasos se siguió a cabalidad con el fin de asegurar una calidad en el proceso mismo y en el producto final.

Guión del Proceso (PSP0)

Propósito	Para guiar el desarrollo de programas a nivel de módulo
Criterios de Entrada	<ul style="list-style-type: none"> – Descripción del Problema – Formulario del Resumen del Plan del Proyecto PSP0.1 – Log de Registro de Tiempos y Defectos – Estándares de Tipo de Defecto, Codificación, y conteo del tamaño – Cronómetro

Paso	Actividades	Descripción
1	Planeación	<ul style="list-style-type: none"> – Producir u obtener listado de requerimientos. – Estimar el tiempo requerido de desarrollo (diseño, codificación, etc.). – Ingresar la información del Plan en el formulario Resumen del Plan de Proyecto. – Completar el Log de Registro de Tiempo.
2	Desarrollo	<ul style="list-style-type: none"> – Diseño del programa. – Implementación del diseño. – Compilación del programa, y reparar registrar todos los defectos encontrados. – Probar el programa, arreglar y registrar todos los defectos encontrados. – Completar el Log de Registro del Tiempo.
3	Postmortem	<ul style="list-style-type: none"> – Completar el formulario del Resumen del Plan de Proyecto con la información de tiempos, defectos y tamaños.

Criterios de Salida	<ul style="list-style-type: none"> – Un programa probado a profundidad – Formulario Resumen del Plan de Proyecto con los datos actuales y estimados. – Log de Registro de Tiempos y Defectos completado
----------------------------	--

1.1. Log de registro de tiempos y defectos

El registro de los logs de registro de tiempos y defectos se llevó a cabo en hojas de cálculo.

1.2. Estándares

PSP tiene unos estándares ya definidos, entre ellos el estándar de tipo de defectos y de conteo.

Estándar de tipo de defectos		
Tipo	Nombre	Descripción
10	Documentación	Comentarios, mensajes
20	Sintaxis	Ortografía, puntuación, formato de las instrucciones
30	Construir, paquetes	Gestión del cambio, librerías, control de versión
40	Asignación	Declaración, nombres duplicados, ámbito, límites
50	Interfaz	Llamadas a procedimientos y referencias, E/S, formatos de usuario
60	Chequeo	Mensajes de error, chequeos inadecuados
70	Datos	Estructura, contenido
80	Función	Lógica, punteros, bucles, recursión, computación, defectos de la función
90	Sistema	Configuración, temporización, memoria
100	Entorno	Diseño, compilación, pruebas y otros problemas que soporta el sistema

Tabla 1. Estándar de defectos

Para el estándar de conteo se usaron Líneas de código (Lines Of Code - LOC)

Para el estándar de codificación se utilizaron los estándares PEP8 [19], un estándar de Python y ECMAScript 6 [20], un estándar de Javascript.

1.3. Guión del proceso de Planeación

Para la fase de planeación se utilizó el siguiente guión:

Guión de la Planeación (PSP0)

Propósito	Guiar el proceso de Planificación del PSP
Criterios de Entrada	<ul style="list-style-type: none"> – Descripción del problema – Formulario resumen del Plan del Proyecto PSP0.1 – Log de Registro del Tiempo

Paso	Actividades	Descripción
1	Requisitos del programa	<ul style="list-style-type: none"> – Producir u obtener una declaración de requisitos para el programa – Asegurar que la declaración de requisitos es claro y no ambiguo
2	Estimación de tamaño	<ul style="list-style-type: none"> – Realizar la mejor estimación del tamaño añadido y modificado del programa – Ingresar información del tamaño planificado en el Formulario Resumen del Plan de Proyecto.
3	Estimación de recursos	<ul style="list-style-type: none"> – Realizar la mejor estimación del tiempo requerido para el desarrollo del programa – Ingresar información del Tiempo planificado en el formulario resumen del Plan del Proyecto – Usando el % a la fecha del programa desarrollado más recientemente, distribuir el tiempo de desarrollo sobre las fases del proyecto planificado.

Criterios de Salida	<ul style="list-style-type: none"> – Declaración de los requisitos documentados – Formulario resumen del plan de proyecto completado con información del tamaño del programa estimado y tiempo de desarrollo – Log de registro de tiempo completado
----------------------------	--

** Verifique que ha cumplido con las condiciones de salida de la fase de planificación.

El guión de planeación también recibió como criterio de entrada la descripción del problema. El formulario de resumen del proyecto que es un formulario donde se llevaron los tiempos planeados por fase (planeación, diseño, desarrollo, pruebas, etc.), los tiempos reales y los tiempos a la fecha, también se llevaron los defectos inyectados y removidos por fase.

El log de registro de tiempos se usó para anotar los tiempos reales de las actividades hechas en cada fase, el log de registro es mucho más detallado con fechas, tiempos, tiempos de interrupción y comentarios de la actividad.

Para los requisitos de la aplicación web se hicieron encuestas a los estudiantes de la Universidad Tecnológica de Pereira, dadas las características de la aplicación también se definieron requerimientos no funcionales.

2. Requisitos del programa

2.1. Recopilación de datos

Con el fin de construir una solución de software de acuerdo a las necesidades de las personas y particularmente de acuerdo a la de los estudiantes de la Universidad Tecnológica de Pereira se realizaron 15 encuestas a estudiantes de diferentes programas de la Universidad.

A continuación se relacionan las preguntas realizadas a los estudiantes de la Universidad con respecto al uso de Banco del Tiempo, se les hizo una pregunta cerrada y dos preguntas abiertas a cada uno.

Antes de empezar la entrevista se le explicó a cada estudiante de manera individual en qué consistía un Banco del Tiempo, que funcionalidades estaban ya planteadas y cuál era el objetivo del mismo.

Estudiante \ Pregunta	¿Usaría la aplicación del Banco del Tiempo Online para intercambiar servicios?	¿Qué tipo de servicios intercambiaría en la aplicación del Banco del Tiempo Online?	¿Qué funcionalidad adicional le añadiría a la aplicación del Banco del Tiempo Online?
1	Si	Clases y prácticas de medicina, clases de literatura y ciencias sociales	Intercambios grupales o con instituciones
2	Si	Intercambios de conocimiento en astronomía, medicina, clases de guitarra, primeros auxilios, rescate, conocimiento sobre árboles, plantas y recursos naturales.	Perfiles con información privada, políticas de privacidad
3	Si	NS/NR	No se que mas podria tener

4	Si	Conocimientos de primeros auxilios e idiomas	Protección de la información personal de las personas registradas
5	Si	NS/NR	Calificar usuarios, para evitar hacer intercambios con gente malintencionada.
6	Si	Clases de natación a cambio de clases de nutrición	NS/NR
7	Si	Intercambiaría clases de manualidades y electrónica	Tener un sistema de referidos, que cada por cada persona que se invite se obtenga tiempo a cambio.
8	Si	Cuidado de personas o enseñanza	Calificación de usuarios como lo hace UBER con una descripción de cómo fue el servicio prestado
9	Si	NS/NR	NS/NR
10	Si	NS/NR	Que cada usuario tenga un perfil interno, considerar realizar una app
11	Si	NS/NR	Así me parece bien
12	Si	Darí tutorías de ciencias básicas (matemáticas, física, etc.) a cambio de servicios de aseo en casa o almuerzos.	Que se pueda destacar mis propuestas a personas que estén buscando lo que yo estoy ofreciendo, permitir que el usuario seleccione qué servicios quiere que le presten al momento de hacer el registro para que las búsquedas se enfoquen en eso
13	Si	NS/NR	NS/NR
14	Si	Clases de francés y manualidades, intercambios culturales	Comprobar usuarios antes de aprobarlos y poder reportar personas que no le estén dando el uso adecuado al sistema, intercambio de bienes en buen estado (trueques)
15	Si	NS/NR	No se me ocurre nada

Tabla 2. Respuestas de las encuestas

De la pregunta “Usaría la aplicación del Banco del Tiempo Online para intercambiar servicios?” el 100% de las respuestas fueron afirmativas, es decir, que los

estudiantes utilizarían el sistema.

De la pregunta “¿Qué tipo de servicios intercambiaría en la aplicación del Banco del Tiempo Online?” el 100% de las personas están interesadas en realizar intercambios de conocimiento, solo una persona mencionó cuidado de personas.

De la pregunta “Qué funcionalidad adicional le añadiría a la aplicación del Banco del Tiempo Online?” 6 personas (40%) no sabían qué funcionalidad adicional podría tener el sistema, de las 11 personas restantes, 3 mencionaron temas relacionados con protección de información, privacidad y perfiles de usuario, también 2 personas mencionaron calificación de los servicios prestados.

Algunas personas mencionaron en las preguntas abiertas el tema de trueques (intercambio de bienes), funcionalidad con más complejidad que será tomada en cuenta para una segunda fase del proyecto.

2.2. Especificación de requerimientos

Especificación de requerimientos de acuerdo a la información recabada de las encuestas a los estudiantes de la universidad y acordes a una solución al planteamiento del problema.

- La aplicación web debe poder ser accedida desde un computador o dispositivo móvil con acceso a internet.
- La aplicación web debe tener un formulario de registro con nombre de usuario, correo electrónico y contraseña accesible para cualquier persona o entidad.
- La aplicación web debe mostrar un perfil de usuario accesible para cualquier usuario de internet, mostrando información básica del perfil como descripción del usuario, habilidades y calificaciones, pero ocultando información confidencial como número de horas, teléfono, correo electrónico.
- La aplicación del banco del tiempo debe permitir generar a cualquier usuario

o entidad registrada hacer una propuesta de algún servicio que requiera.

- La aplicación debe permitir a cualquier perfil de usuario registrado buscar propuestas realizadas por otros perfiles de acuerdo a sus habilidades y ubicación
- La aplicación web debe permitir a cualquier usuario o entidad registrada hacer una oferta a alguna propuesta publicada.
- La aplicación debe permitir realizar transferencias de horas por una propuesta u oferta realizada entre 2 perfiles.
- La aplicación del banco del tiempo debe guardar de manera íntegra la cantidad de horas de cada perfil registrado y las transacciones por cada propuesta u oferta en la que intervenga el perfil.
- La aplicación web deberá permitir a un perfil que recibió un servicio de otro perfil hacer una calificación de 1 a 5 y viceversa
- La aplicación debe permitir a los usuarios recibir notificaciones de otros usuarios o del mismo sistema (mensajería).
- El sistema contará con unos términos y condiciones que deberán ser aceptados al realizar un registro y unas políticas de privacidad

Las demás sugerencias realizadas por los estudiantes como comprobación de usuarios, intercambios de bienes, destacar propuestas y sistema de referidos serán realizadas en una siguiente fase del proyecto.

2.3. Alcances

La aplicación tendrá los siguientes módulos:

2.3.1. Módulo de usuarios

Este módulo abarca como tal la administración de perfiles dentro de la aplicación web del banco del tiempo, como por ejemplo: registro de usuarios, autenticación de usuarios, actualización de datos de usuario.

2.3.2. Módulo de intercambios

Este módulo contiene como tal la administración de los registros de servicios y postulaciones de intercambios entre usuarios.

2.3.3. Módulo de notificaciones

Este módulo se encargará de notificar a los usuarios del sistema en caso de que haya alguna postulación a un servicio ofrecido o viceversa.

2.3.4. Módulo de calificaciones

Módulo que permitirá a un usuario calificar a otro al prestar o recibir un servicio y viceversa.

2.4. Cronograma

Actividad	Objetivos	Meses					
		1	2	3	4	5	6
A	Realizar entrevistas a una muestra de 15 personas de la Universidad Tecnológica de Pereira	■					
C	Definir los requerimientos de software y de hardware para la aplicación	■					
F	Realizar las etapas de análisis, diseño y diagramación de acuerdo a los requerimientos definidos anteriormente	■	■				
G	Implementación módulo de usuarios (registro, cuentas, etc.)		■	■			
H	implementación módulo de intercambios (registros de intercambios, postulación a servicios, notificaciones)			■	■		
I	Implementación módulo de notificaciones (mensajes de postulaciones y servicios)				■		
J	Integración de todo el sistema				■	■	
L	Realización de pruebas de toda la aplicación						■

Tabla 3. Cronograma

2.5. Arquitectura de Hardware y Software

En vista de que la aplicación debe ser en línea, disponible a través de internet para cualquier persona se realizó una aplicación web.

Para el desarrollo de software se usaron las siguientes tecnologías libres y de código abierto.

Django: es un framework web de alto nivel hecho en Python, se enfoca en proveer a los desarrolladores una herramienta para construir aplicaciones web más rápido y con menos código, además toma en cuenta solucionar desde el principio problemas de seguridad y permite escalar las aplicaciones fácilmente [21].

HTML5: Lenguaje de marcado para la estructura y presentación de contenidos de la World Wide Web [22]

CSS3: CSS es un lenguaje para definir el estilo o la apariencia de las páginas web, escritas con HTML o de los documentos XML. CSS se creó para separar el contenido de la forma, a la vez que permite a los diseñadores mantener un control mucho más preciso sobre la apariencia de las páginas. [23]

Javascript: Javascript es un lenguaje de programación que permite a los desarrolladores crear acciones en sus páginas web. [24]

Reactjs: Es una biblioteca escrita en JavaScript, desarrollada en Facebook para facilitar la creación de componentes interactivos, reutilizables, para interfaces de usuario. [25]

Postgres: PostgreSQL es un potente sistema de base de datos objeto-relacional de código abierto. Cuenta con más de 15 años de desarrollo activo y una arquitectura probada que se ha ganado una sólida reputación de fiabilidad e integridad de datos. [26]

Gitlab: Servicio de control de versiones, se usará para alojar todo el código desarrollado, entre sus características se destacan seguimiento de las mejoras, nuevas funcionalidades o errores del software, facilidad del uso y rendimiento.[27]

Para soportar la aplicación se utilizó la siguiente arquitectura de Hardware:

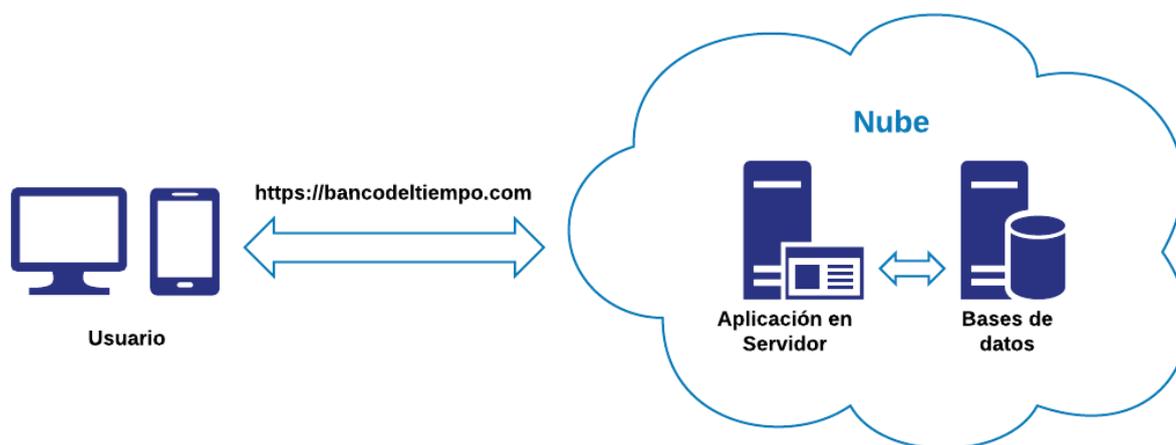


Imagen 1. Arquitectura de hardware

La nube en este caso fue básicamente un Servidor Virtual Privado (VPS), allí se alojó la aplicación web, el usuario accede a la misma desde un computador o dispositivo móvil a través de internet.

3. Estimación de tiempos

Uno de los objetivos de PSP es que se haga una mejora continua del proceso de desarrollo, para esto se utilizan datos históricos de tamaños (en líneas de código, hojas, etc.) y tiempos de cada una de las fases, datos históricos con los que no se contaron por ser el primer desarrollo al que se aplicaba esta metodología y que de haber tenido se habría hecho una mejor estimación, ya que PSP cuenta con algunos métodos desarrollados para estimar.

PSP hace énfasis en que es más fácil estimar el desarrollo de un programa dividiéndolo en varias partes, los módulos se dividieron en partes más pequeñas para hacer una mejor estimación.

Fase	Actividad	Tareas	Tiempo estimado (horas)
Planeación		Planteamiento del problema	2:00:00
		Justificación	1:30:00
		Objetivos	1:00:00
		Conceptos	2:00:00
		Entrevistas a los estudiantes	4:30:00
		Especificación de requerimientos del software	2:00:00
		Alcances de la aplicación web	1:30:00
		Arquitectura de hardware y software	2:00:00
		Estimación de tiempos	3:00:00
		Anotación de tiempos en el log	1:00:00
		Cronograma	2:00:00
Desarrollo	Diseño	Modelo relacional	4:00:00
		Especificación modelo relacional	2:00:00
		Diagrama de casos de uso	5:00:00
		Especificación de casos de uso	2:00:00
		Diagramas de secuencia	2:00:00
		Anotación de defectos en el log de defectos	0:30:00
		Anotación de tiempos en el log de tiempos	0:30:00

Revisión del Diseño	Revisión modelo relacional	3:00:00
	Revisión diagramas de casos de uso	3:00:00
	Revisión Diagramas de secuencia	1:00:00
	Anotación de defectos en el log de defectos	0:30:00
	Anotación de tiempos en el log de tiempos	0:30:00
Codificación	Instalación dependencias del proyecto (Django, postgresql)	1:00:00
	Configuración inicial del proyecto	2:00:00
	Implementación estilos css base del proyecto	1:30:00
	Implementación menu, footer generales del usuario	3:00:00
	Implementación de modelo relacional	3:00:00
	Configuración de modelos en el administrador de Django	1:00:00
	Implementación serializadores (Para construir APIs)	1:30:00
	Implementación registro de usuarios y entidades	2:00:00
	Implementación login de usuarios	2:00:00
	Implementación vista de perfil de un usuario	4:00:00
	Implementación opción de actualizar usuario	2:30:00
	Implementación mensajes de alerta	1:00:00
	Implementación opción de cambiar contraseña	1:30:00
	Implementación creación de propuestas	2:00:00
	Implementación vista de propuesta como página	2:00:00
	Implementación actualización de propuestas	2:30:00
	Implementación buscador y filtro (lugar, habilidades, horas) de propuestas	5:00:00
	Implementación opción de hacer ofertas a las propuestas	3:00:00
	Implementación opción de editar ofertas	2:00:00
	Implementación opción de aceptar ofertas a propuestas	1:30:00
	Implementación opción de cancelar ofertas y propuestas	1:30:00
	Implementación finalización propuestas y transferencias de horas	3:00:00
	Implementación envío de mensajes entre usuarios	4:00:00
	Implementación dashboard de notificaciones de mensajes	3:00:00
	Integración envío de mensajes al recibir, aceptar, cancelar postulaciones	1:30:00
	Implementación calificaciones entre usuarios	4:00:00
	Integración calificaciones al perfil de usuario público	1:00:00
Implementación términos y condiciones del proyecto y aceptación en el registro	4:00:00	
Anotación de defectos en el log de defectos	1:00:00	

	Anotación de tiempos en el log de tiempos	1:00:00
Revisión de codificación	Revisión de la configuración inicial del proyecto	0:30:00
	Revisión de Implementación estilos css base del proyecto	0:30:00
	Revisión de Implementación menu, footer generales del usuario	1:00:00
	Revisión de Implementación de modelo relacional	1:00:00
	Revisión de configuración de modelos en el administrador de Django	0:20:00
	Revisión de Implementación serializadores	0:30:00
	Revisión de Implementación registro de usuarios y entidades	0:30:00
	Revisión de Implementación login de usuarios	0:30:00
	Revisión de Implementación vista de perfil de un usuario	1:00:00
	Revisión de Implementación opción de actualizar usuario	0:30:00
	Revisión de Implementación mensajes de alerta	0:20:00
	Revisión de Implementación opción de cambiar contraseña	0:20:00
	Revisión de Implementación creación de propuestas	1:00:00
	Revisión de Implementación vista de propuesta como página	0:40:00
	Revisión de Implementación actualización de propuestas	0:30:00
	Revisión de Implementación buscador y filtro (lugar, habilidades, horas) de propuestas	1:30:00
	Revisión de Implementación opción de hacer ofertas a las propuestas	1:00:00
	Revisión de Implementación opción de editar ofertas	0:40:00
	Revisión de Implementación opción de aceptar ofertas a propuestas	0:30:00
	Revisión de Implementación opción de cancelar ofertas y propuestas	0:30:00
	Revisión de Implementación finalización propuestas y transferencias de horas	1:00:00
	Revisión de Implementación envío de mensajes entre usuarios	1:00:00
	Revisión de Implementación dashboard de notificaciones de mensajes	1:00:00
	Revisión de Integración envío de mensajes al recibir, aceptar, cancelar postulaciones	0:30:00
Revisión de Implementación calificaciones entre usuarios	1:00:00	
Revisión de Integración calificaciones al perfil de usuario público	0:20:00	

		Revisión de Implementación términos y condiciones del proyecto y aceptación en el registro	1:00:00
		Anotación de defectos en el log de defectos	1:00:00
		Anotación de tiempos en el log de tiempos	1:00:00
	Compilación	Generación de archivos preprocesados de python	1:00:00
		Solucionar errores que impidan la generación de los archivos preprocesados	10:00:00
		Anotación de defectos en el log de defectos	2:00:00
		Anotación de tiempos en el log de tiempos	1:00:00
	Pruebas	Realizar pruebas del funcionamiento del módulo hasta no encontrar defectos	12:00:00
		Corrección de los defectos encontrados	4:00:00
		Anotación de defectos en el log de defectos	2:00:00
		Anotación de tiempos en el log de tiempos	1:00:00
PostMortem	Completar el formulario del plan del proyecto (tiempos, defectos, etc)	2:00:00	
	Realizar Cálculos estadísticos para saber la calidad del proceso y del producto	6:00:00	
	Realizar Plan de Mejora del Proceso	2:00:00	

Tabla 4. Estimación de tiempos

También se anotaron los tiempos reales tomados por cada actividad (Se mostrarán los totales más adelante), el propósito de esta información es mejorar la estimación de tiempos que tomaría un próximo desarrollo de software.

En vista de que el proceso de codificación requiere de tener módulos terminados, probados y que estén funcionando bien para poder continuar con el desarrollo de nuevos módulos, PSP permite hacer iteraciones sobre las fases de codificación, revisión de codificación, compilación y pruebas, por ejemplo:

1. Configuración inicial del proyecto
2. Revisión de la Configuración inicial del proyecto
3. Compilación de la configuración inicial del proyecto
4. Pruebas de la configuración inicial del proyecto.

Al terminar las pruebas se continúa con el siguiente ítem del proceso de desarrollo, que dentro de la planeación realizada sería: Implementación estilos css base del proyecto.

VI Desarrollo

PSP tiene también un guión para el desarrollo:

Guión de la Desarrollo (PSP0)

Propósito	Para guiar el desarrollo de pequeños programas
Criterios de Entrada	<ul style="list-style-type: none"> – Declaración de requisitos – Formulario resumen del Plan del Proyecto con el tiempo estimado para el desarrollo del programa – Logs de registro de Tiempo y Defectos – Estándar del tipo de defectos

Paso	Actividades	Descripción
1	Diseño	<ul style="list-style-type: none"> – Revisar los requisitos y producir un diseño para lograrlos. – Registrar en el log de registro de Defectos cualquier defecto de requisitos encontrado. – Registrar el tiempo en el log de registro de Tiempos.
2	Codificación	<ul style="list-style-type: none"> – Implementar el diseño. – Registrar en el log de Registro de Defectos cualquier defecto de requisitos o de diseño encontrados. – Registrar el tiempo en el log de Registro de Tiempo.
3	Compilación	<ul style="list-style-type: none"> – Compilar el programa hasta que se encuentre libre de errores. – Reparar todos los defectos encontrados. – Registrar los defectos en el log de Registro de Defectos. – Registrar el tiempo en el log de Registro de Tiempos.
4	Pruebas	<ul style="list-style-type: none"> – Probar hasta que las pruebas se ejecuten sin errores. – Reparar todos los defectos encontrados. – Registrar los defectos en el log de Registro de Defectos. – Registrar el tiempo en el log de Registro de Tiempos.

Criterios de Salida	<ul style="list-style-type: none"> – Un programa probado a profundidad – Logs de Registro de Defectos y de Tiempos completado.
----------------------------	--

** Verifique que ha cumplido con las condiciones de salida de la fase de planificación.

Los criterios de entrada ya se habían definido anteriormente en la sección de Planeación. PSP permite un desarrollo circular, primero se definió todo el diseño de la aplicación, y la codificación, diseño y pruebas se realizaron por cada módulo definido.

7. Diseño del Software

7.1. Modelo relacional

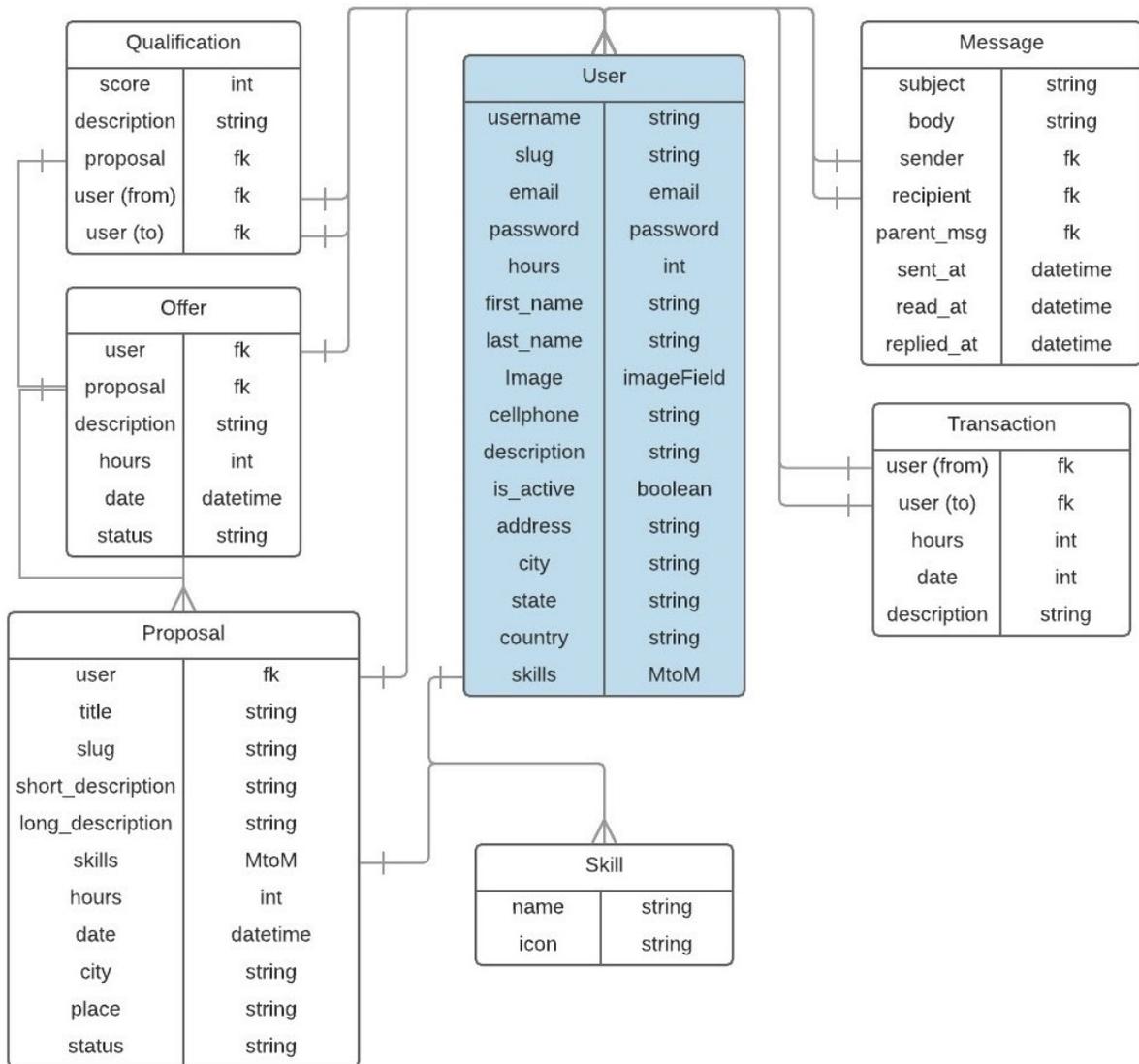


Imagen 2. Modelo Relacional

7.2. Especificación del modelo

El modelo relacional se hizo de acuerdo al framework de desarrollo Django, que facilita un poco las relaciones muchos a muchos con otras entidades (MtoM o Many to Many)

El modelo de propuestas (Proposal) tiene un campo llamado estatus (status) que se utilizó para definir el estado actual de la propuesta, puede tener los siguientes valores: publicada, en progreso, cancelada y terminada.

El modelo de Ofertas (Offer) tiene también un campo llamado estatus (status), puede tener los siguientes valores: publicada, aceptada, declinada, terminada.

7.3. Casos de uso

Casos de uso de los módulos definidos y su diagrama

7.3.1. Módulo de Usuarios

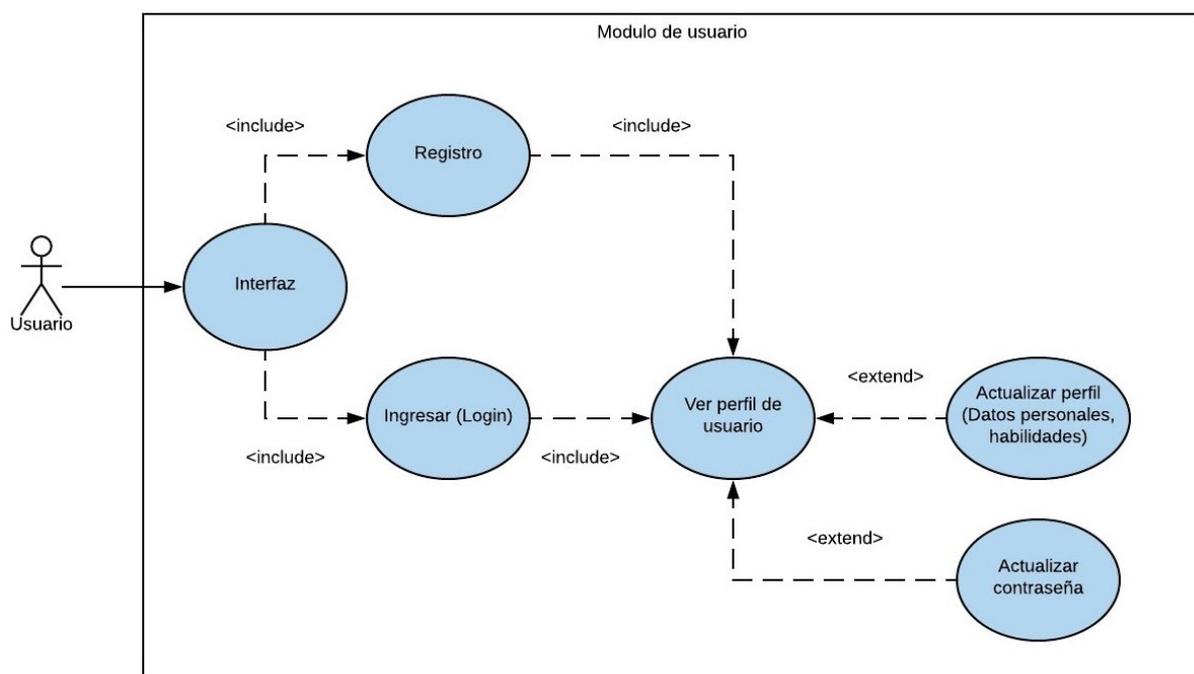


Imagen 3. Módulo de usuario

Caso de uso	CU-01: Registro
Actores	Usuario
Propósito	Este caso de uso permite a los usuarios registrarse en el sistema
Resumen	El usuario ingresa a la página de inicio, en el menú hay un botón de registro que lo lleva al formulario de registro
Precondición	Ninguna

Tipo	Primario
Clase	Esencial
Curso Normal de eventos	
Acciones de los actores	Respuesta del sistema
1. El usuario ingresa al sitio web e ingresa en el menú a registro	
	2. El sistema muestra el formulario de registro
3. El usuario completa el formulario de registro	
	4. El usuario queda registrado
Curso alterno	
4. Si ya hay un usuario registrado con los mismos datos (correo) o algún error de validación se mostrará un mensaje al usuario	

Caso de uso	CU-02: Ingresar (Login)
Actores	Usuario
Propósito	Este caso de uso permite a los usuarios ingresar al sistema
Resumen	El usuario ingresa a la página de inicio, en el menú hay un botón de ingreso o login que lo lleva a un formulario
Precondición	Ninguna
Tipo	Primario
Clase	Esencial
Curso Normal de eventos	
Acciones de los actores	Respuesta del sistema
1. El usuario ingresa al sitio web e ingresa en el menú a ingresar	
	2. El sistema muestra el formulario de ingreso
3. El usuario completa el formulario de ingreso con sus datos	
	4. El usuario ingresa al sitio
Curso alterno	
4. Si ya hay algún error de validación se mostrará un mensaje al usuario	

Caso de uso	CU-03: Ver perfil de usuario
Actores	Usuario

Propósito	Este caso de uso permite a los usuarios ver su perfil
Resumen	El usuario ingresa a la página, se registra o ingresa y puede ver su perfil
Precondición	CU-01 o CU-02
Tipo	Primario
Clase	Esencial
Curso Normal de eventos	
Acciones de los actores	Respuesta del sistema
1. El usuario ingresa en el menú a ver su perfil	
	2. El sistema muestra el perfil

Caso de uso	CU-04: Actualizar perfil
Actores	Usuario
Propósito	Este caso de uso permite a los usuarios actualizar sus datos de perfil
Resumen	El usuario ingresa a la página de inicio, debe registrarse o ingresar, ir a su perfil y luego ir a actualizar perfil
Precondición	CU-03
Tipo	Primario
Clase	Esencial
Curso Normal de eventos	
Acciones de los actores	Respuesta del sistema
1. En el perfil el usuario ingresa a actualizar su perfil	
	2. El sistema muestra un formulario con sus datos
3. El usuario completa el formulario y actualiza sus datos	
	4. El sistema actualiza el perfil
Curso alterno	
4. Si hay algún error de validación se mostrará un mensaje al usuario	

Caso de uso	CU-05: Actualizar contraseña
Actores	Usuario
Propósito	Este caso de uso permite a los usuarios

	actualizar su contraseña
Resumen	El usuario ingresa a la página de inicio, debe registrarse o ingresar, ir a su perfil y luego ir a actualizar contraseña
Precondición	CU-03
Tipo	Primario
Clase	Esencial
Curso Normal de eventos	
Acciones de los actores	Respuesta del sistema
1. En el perfil el usuario ingresa a actualizar contraseña	
	2. El sistema muestra un formulario para que ingrese su contraseña actual y la nueva
3. El usuario completa el formulario	
	4. El sistema actualiza la contraseña
Curso alternativo	
4. Si hay algún error de validación se mostrará un mensaje al usuario	

7.3.2 Módulo de intercambios

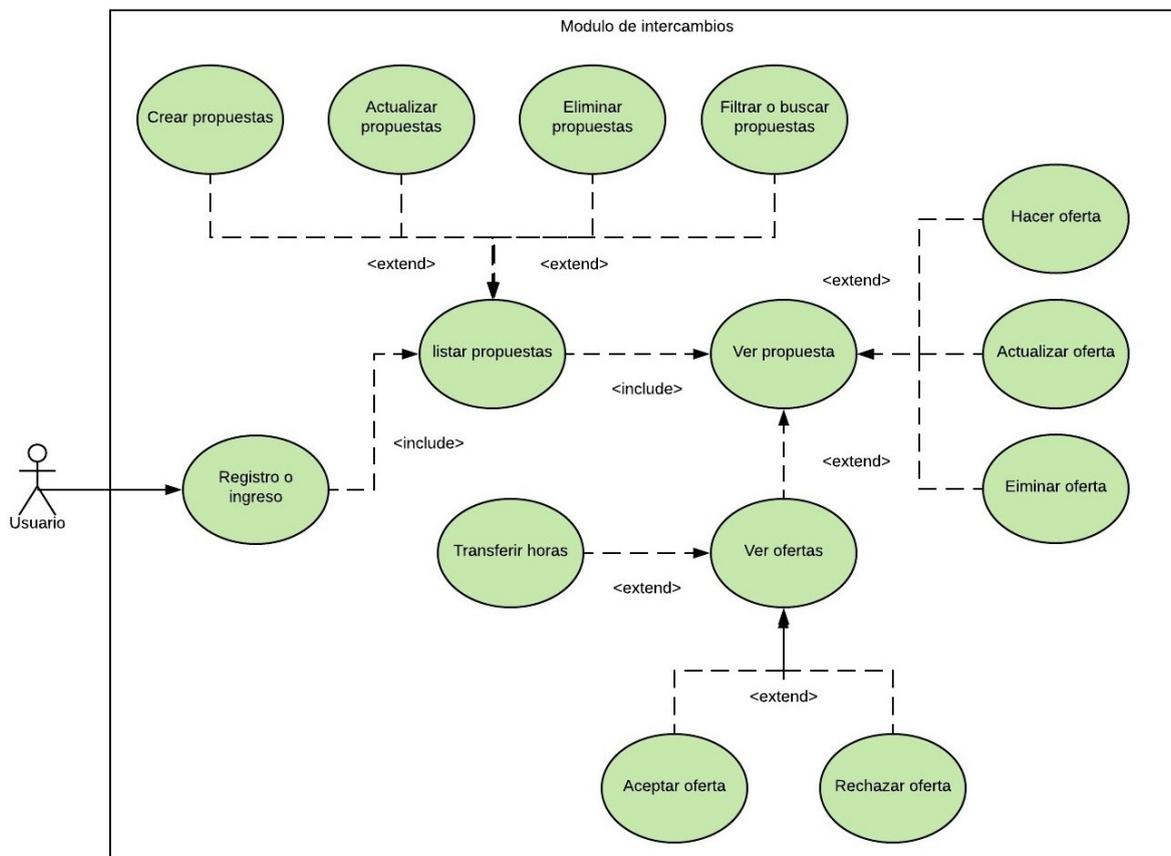


Imagen 4. Módulo de intercambios

Caso de uso	CU-06: Listar propuestas
Actores	Usuario
Propósito	Este caso de uso permite a los usuarios ver las propuestas creadas por ellos o por otros usuarios
Resumen	El usuario debe registrarse o ingresar, ir luego en el menú a ver propuestas
Precondición	CU-01 o CU-02
Tipo	Primario
Clase	Esencial
Curso Normal de eventos	
Acciones de los actores	Respuesta del sistema
1. En el menú ingresar a listar propuestas	
	2. El sistema lista las propuestas creadas por él y filtra otras creadas por otros

	usuarios
--	----------

Caso de uso	CU-07: Crear propuestas
Actores	Usuario
Propósito	Este caso de uso permite al usuario crear propuestas
Resumen	El usuario debe registrarse o ingresar, ir luego en el menú a ver propuestas y luego a crear propuestas
Precondición	CU-06
Tipo	Primario
Clase	Esencial
Curso Normal de eventos	
Acciones de los actores	Respuesta del sistema
1. En el listado de propuestas ir a crear propuesta	
	2. El sistema muestra un formulario para crear la nueva propuesta
3. Completar el formulario	
	4. La propuesta queda creada
Curso alterno	
4. En caso de algún error de validación se mostrará mensaje al usuario	

Caso de uso	CU-08: Actualizar propuestas
Actores	Usuario
Propósito	Este caso de uso permite al usuario actualizar propuestas ya creadas
Resumen	El usuario debe registrarse o ingresar, ir luego en el menú a ver propuestas y luego a actualizar propuesta
Precondición	CU-06
Tipo	Primario
Clase	Esencial
Curso Normal de eventos	
Acciones de los actores	Respuesta del sistema
1. En el listado de propuestas ir a actualizar alguna de sus propuestas	
	2. El sistema muestra un formulario para

	actualizar la propuesta creada
3. Completar el formulario	
	4. La propuesta queda actualizada
Curso alterno	
4. En caso de algún error de validación se mostrará mensaje al usuario	

Caso de uso	CU-09: Eliminar propuestas
Actores	Usuario
Propósito	Este caso de uso permite al usuario eliminar sus propuestas ya creadas
Resumen	El usuario debe registrarse o ingresar, ir luego en el menú a ver propuestas y luego ir a eliminar propuesta
Precondición	CU-06
Tipo	Primario
Clase	Esencial
Curso Normal de eventos	
Acciones de los actores	Respuesta del sistema
1. En el listado de propuestas seleccionar eliminar alguna de sus propuestas	
	2. El sistema pregunta si está seguro de la acción
3. Confirma la acción de eliminar	
	4. La propuesta queda eliminada
Curso alterno	
4. En caso de no confirmar la acción la propuesta no se elimina	

Caso de uso	CU-10: Buscar o filtrar propuestas
Actores	Usuario
Propósito	Este caso de uso permite al usuario Buscar o filtrar propuestas de otros usuarios
Resumen	El usuario debe registrarse o ingresar, ir luego en el menú a ver propuestas y luego puede buscar o filtrar propuestas
Precondición	CU-06
Tipo	Primario

Clase	Esencial
Curso Normal de eventos	
Acciones de los actores	Respuesta del sistema
1. En el listado de propuestas buscar o filtrar propuestas	
	2. El sistema retorna el un listado de propuestas de acuerdo a lo ingresado por el usuario
Curso alterno	
2. En caso de no encontrar resultados se mostrará un mensaje al usuario de propuestas no encontradas	

Caso de uso	CU-11: Ver propuesta en detalle
Actores	Usuario
Propósito	Este caso de uso permite al usuario ver una propuesta en detalle
Resumen	El usuario debe registrarse o ingresar, ir luego en el menú a ver propuestas y luego ver propuesta en detalle
Precondición	CU-06
Tipo	Primario
Clase	Esencial
Curso Normal de eventos	
Acciones de los actores	Respuesta del sistema
1. En el listado de propuestas seleccionar ver propuesta en detalle	
	2. El sistema muestra la propuesta en detalle

Caso de uso	CU-12: Hacer oferta a propuesta
Actores	Usuario
Propósito	Este caso de uso permite realizar una oferta a una propuesta
Resumen	El usuario debe registrarse o ingresar, ir luego en el menú a ver propuestas y luego ver propuesta en detalle y luego hacer una oferta
Precondición	CU-11
Tipo	Primario

Clase	Esencial
Curso Normal de eventos	
Acciones de los actores	Respuesta del sistema
1. En la propuesta en detalle hacer una oferta	
	2. El sistema muestra un formulario donde se hará la oferta a la propuesta
3. llenar el formulario con los datos de la oferta	
	4. Crear la propuesta a la oferta y avisar al usuario de la propuesta
Curso alterno	
4. En caso de algún error de validación se mostrará un mensaje al usuario	

Caso de uso	CU-13: Actualizar oferta
Actores	Usuario
Propósito	Este caso de uso permite actualizar una oferta realizada a una propuesta
Resumen	Al ver una propuesta en detalle con una oferta ya realizada, actualizar la oferta
Precondición	CU-11
Tipo	Primario
Clase	Esencial
Curso Normal de eventos	
Acciones de los actores	Respuesta del sistema
1. En la propuesta en detalle ir a actualizar oferta ya realizada	
	2. El sistema muestra un formulario donde se actualiza la oferta
3. actualizar los datos de la oferta	
	4. Guarda los datos de la oferta
Curso alterno	
4. En caso de algún error de validación se mostrará un mensaje al usuario	

Caso de uso	CU-14: Eliminar oferta
Actores	Usuario
Propósito	Este caso de uso es para eliminar una oferta realizada a una propuesta

Resumen	Al ver una propuesta en detalle con una oferta ya realizada, eliminar la oferta
Precondición	CU-11
Tipo	Primario
Clase	Esencial
Curso Normal de eventos	
Acciones de los actores	Respuesta del sistema
1. En la propuesta en detalle ir a eliminar oferta	
	2. El sistema muestra un mensaje de confirmación para eliminar la oferta
3. El usuario confirma que desea eliminar la oferta	
	4. La oferta es eliminada
Curso alterno	
4. En caso de no confirmar el eliminar la oferta, esta no será eliminada	

Caso de uso	CU-15: Ver ofertas
Actores	Usuario
Propósito	Este caso de uso es ver todas ofertas realizadas a una propuesta de un usuario
Resumen	Al ver una propuesta en detalle, ver todas las propuestas
Precondición	CU-11
Tipo	Primario
Clase	Esencial
Curso Normal de eventos	
Acciones de los actores	Respuesta del sistema
1. En la propuesta en detalle ir a ver ofertas	
	2. El sistema muestra todas las ofertas realizadas a la propuesta

Caso de uso	CU-16: Aceptar oferta
Actores	Usuario
Propósito	Este caso de uso es para aceptar una oferta realizada a una propuesta
Resumen	Al ver una propuesta en detalle, ver

	todas las propuestas y aceptar una de ellas
Precondición	CU-11
Tipo	Primario
Clase	Esencial
Curso Normal de eventos	
Acciones de los actores	Respuesta del sistema
1. En el listado de ofertas de una propuesta aceptar una.	
	2. El sistema muestra un mensaje de confirmación para aceptar la oferta
3. El usuario confirma que desea aceptar la oferta	
	4. La oferta es aceptada y se envía un mensaje al usuario de la oferta
Curso alterno	
4. En caso de no confirmar la aceptación el sistema no hará nada	

Caso de uso	CU-17: Rechazar oferta
Actores	Usuario
Propósito	Este caso de uso es para rechazar una oferta realizada a una propuesta
Resumen	Al ver una propuesta en detalle, ver todas las propuestas y rechazar las ofertas
Precondición	CU-11
Tipo	Primario
Clase	Esencial
Curso Normal de eventos	
Acciones de los actores	Respuesta del sistema
1. En el listado de ofertas de una propuesta rechazar una oferta.	
	2. El sistema muestra un mensaje de confirmación para rechazar la oferta
3. El usuario confirma que desea rechazar la oferta	
	4. La oferta es rechazada y se envía un mensaje al usuario de la oferta
Curso alterno	
4. En caso de no confirmar el rechazo de la oferta el sistema no hará nada	

Caso de uso	CU-18: Transferir horas
Actores	Usuario
Propósito	Este caso de uso es para transferir horas a una propuesta aceptada
Resumen	Al ser una oferta de una propuesta aceptada se puede transferir las horas acordadas en la misma
Precondición	CU-15
Tipo	Primario
Clase	Esencial
Curso Normal de eventos	
Acciones de los actores	Respuesta del sistema
1. En la propuesta con una oferta de otro usuario aceptada ir a transferir horas	
	2. El sistema muestra un mensaje de confirmación para realizar la transferencia de horas
3. El usuario confirma que desea transferir las horas al usuario de la oferta	
	4. Las horas son transferidas, se notifica al usuario de la oferta.
Curso alterno	
4. En caso de no confirmar la transferencia de las horas el sistema no hará nada	

7.3.3 Módulo de Mensajes

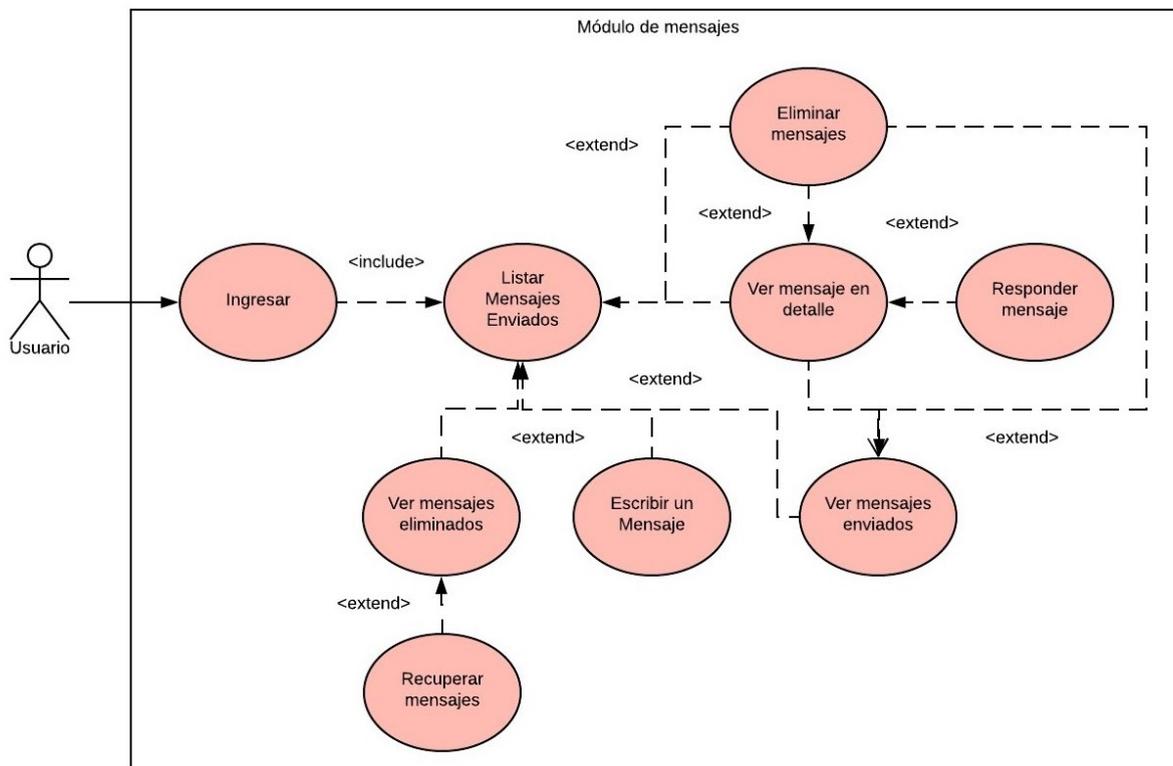


Imagen 5. Módulo de mensajes

Caso de uso	CU-19: Listar mensajes recibidos
Actores	Usuario
Propósito	Este caso de uso muestra los mensajes recibidos a un usuario
Resumen	luego de loguearse el usuario puede ver los mensajes que ha recibido desde el menú
Precondición	CU-01 o CU-02
Tipo	Primario
Clase	Esencial
Curso Normal de eventos	
Acciones de los actores	Respuesta del sistema
1. En el menú ingresar a los mensajes recibidos	
	2. El sistema muestra los mensajes recibidos

Caso de uso	CU-20: Ver mensaje en detalle
Actores	Usuario
Propósito	Este caso de uso muestra en detalle un mensaje recibido o enviado
Resumen	Luego de abrir los mensajes se puede ver en detalle un mensaje
Precondición	CU-19 o CU-23
Tipo	Primario
Clase	Esencial
Curso Normal de eventos	
Acciones de los actores	Respuesta del sistema
1. En el listado de mensajes ver un mensaje en detalle	
	2. El sistema muestra el mensaje en detalle

Caso de uso	CU-21: Responder mensaje
Actores	Usuario
Propósito	Este caso de uso es para responder un mensaje recibido
Resumen	Luego de ver un mensaje en detalle se puede responder el mismo al usuario que lo escribió
Precondición	CU-20
Tipo	Primario
Clase	Esencial
Curso Normal de eventos	
Acciones de los actores	Respuesta del sistema
1. Al abrir un mensaje en detalle ir a responder mensaje	
	2. El sistema muestra un formulario con para ser llenado con la respuesta del mensaje
3. El usuario escribe el mensaje de respuesta y lo envía	
	4. El sistema envía el mensaje
Curso alterno	
4. En caso de error en la validación del formulario el sistema muestra un mensaje	

Caso de uso	CU-22: Eliminar un mensaje
Actores	Usuario
Propósito	Este caso de uso es para eliminar un mensaje
Resumen	Luego de ver un mensaje en detalle o de ver un listado de mensajes se puede eliminar un mensaje
Precondición	CU-19 o CU-20 o CU-23
Tipo	Primario
Clase	Esencial
Curso Normal de eventos	
Acciones de los actores	Respuesta del sistema
1. Al abrir un mensaje o ver un listado de mensajes ir a eliminar mensaje	
	2. El sistema eliminará el mensaje

Caso de uso	CU-23: Ver mensajes enviados
Actores	Usuario
Propósito	Este caso de uso es para ver un listado de mensajes enviados
Resumen	En el submenú de mensajes hay una opción para ver los mensajes enviados
Precondición	CU-19
Tipo	Primario
Clase	Esencial
Curso Normal de eventos	
Acciones de los actores	Respuesta del sistema
1. En el submenú de mensajes ir a mensajes enviados	
	2. El sistema mostrará el listado de mensajes

Caso de uso	CU-24: Escribir un mensaje
Actores	Usuario
Propósito	Este caso de uso es para mostrar un formulario para escribir un mensaje
Resumen	En el submenú de mensajes hay una

	opción para escribir un mensaje
Precondición	CU-19
Tipo	Primario
Clase	Esencial
Curso Normal de eventos	
Acciones de los actores	Respuesta del sistema
1. En el submenú de mensajes ir a escribir mensaje	
	2. El sistema muestra un formulario
3. El usuario debe ingresar los datos del mensaje y el destinatario	
	4. El sistema envía el mensaje
Curso alterno	
4. En caso de error en la validación del formulario el sistema muestra un mensaje	

Caso de uso	CU-25: Ver mensajes eliminados
Actores	Usuario
Propósito	Este caso de uso es para mostrar los mensajes eliminados
Resumen	En el submenú de mensajes hay una opción para ver mensajes eliminados
Precondición	CU-19
Tipo	Primario
Clase	Esencial
Curso Normal de eventos	
Acciones de los actores	Respuesta del sistema
1. En el submenú de mensajes ir a mensajes eliminados	
	2. El sistema muestra los mensajes eliminados

Caso de uso	CU-26: Recuperar mensajes eliminados
Actores	Usuario
Propósito	Este caso de uso es para mostrar recuperar los mensajes eliminados
Resumen	Al ingresar en el listado de mensajes eliminados se puede recuperar un mensaje

Precondición	CU-25
Tipo	Primario
Clase	Esencial
Curso Normal de eventos	
Acciones de los actores	Respuesta del sistema
1. En el listado de mensajes eliminados ir a recuperar mensaje	
	2. El sistema se recupera y vuelve al listado de mensajes recibidos o enviados

7.3.4 Módulo de calificaciones

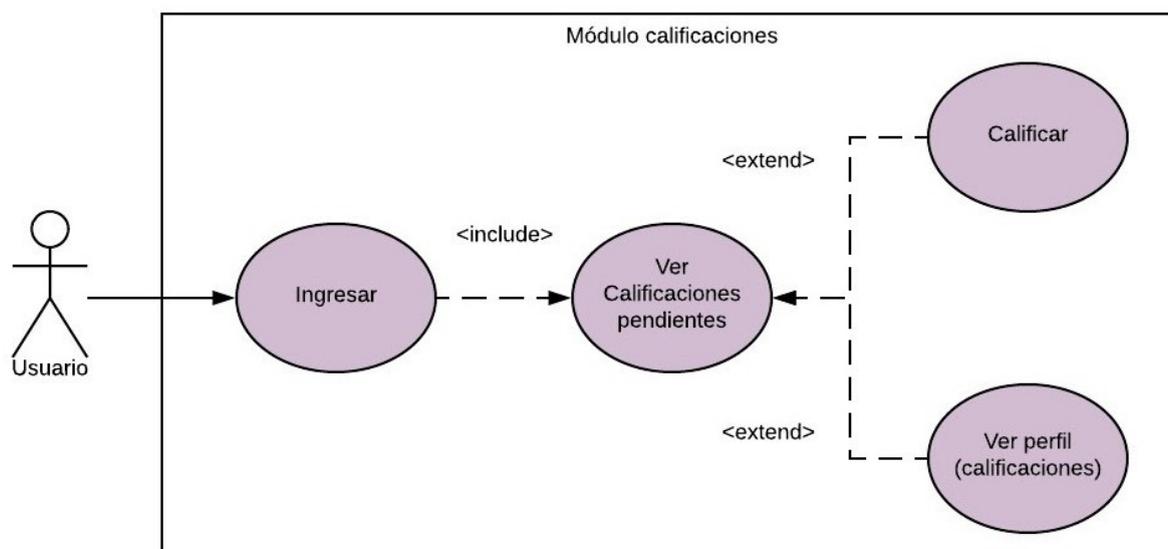


Imagen 6. Módulo de calificaciones

Caso de uso	CU-27: Ver calificaciones pendientes
Actores	Usuario
Propósito	Este caso de uso permite que un usuario vea si tiene calificaciones pendientes por servicios prestados o recibidos
Resumen	Luego de ingresar al sistema un usuario puede ver si tiene calificaciones de usuario pendientes
Precondición	CU-01 o CU-02
Tipo	Primario
Clase	Esencial
Curso Normal de eventos	
Acciones de los actores	Respuesta del sistema
1. En el menú ir a ver calificaciones pendientes	
	2. El sistema muestra las calificaciones pendientes del usuario

Caso de uso	CU-28: Calificar usuario
Actores	Usuario

Propósito	Este caso de uso permite que un usuario pueda calificar a otro por un servicio prestado y viceversa
Resumen	Luego de ingresar al sistema un usuario puede ver si tiene calificaciones de usuario pendientes y calificarlos
Precondición	CU-27
Tipo	Primario
Clase	Esencial
Curso Normal de eventos	
Acciones de los actores	Respuesta del sistema
1. En el listado de calificaciones pendientes ir a calificar usuario	
	2. El sistema muestra un formulario para la calificación
3. El usuario califica a otro usuario	
	4. El sistema guarda la calificación
Curso alterno	
4. En caso de error en la validación del formulario el sistema muestra un mensaje	

Caso de uso	CU-29: Ver calificaciones
Actores	Usuario
Propósito	Este caso de uso permite que un usuario vea como es el perfil de un usuario y su calificación
Resumen	En el listado de calificaciones pendientes el usuario puede ver las calificaciones que los usuarios tienen
Precondición	CU-27
Tipo	Primario
Clase	Esencial
Curso Normal de eventos	
Acciones de los actores	Respuesta del sistema
1. En el listado de calificaciones pendientes ir al perfil del usuario	
	2. El sistema en el perfil del usuario sus calificaciones

7.4. Diagramas de Secuencia

Se realizaron los diagramas de secuencia que muestran de la mejor manera como funciona la aplicación web utilizando el patrón de diseño Modelo Vista Plantilla (Template) o MVT.

Los diagramas de secuencia realizados abarcan varios casos de uso, todos inician asumiendo que ya hay un usuario registrado que ingresa exitosamente (con un registro de usuario también se puede llegar al mismo lugar).

7.4.1 Caso de uso 4, Actualizar usuario

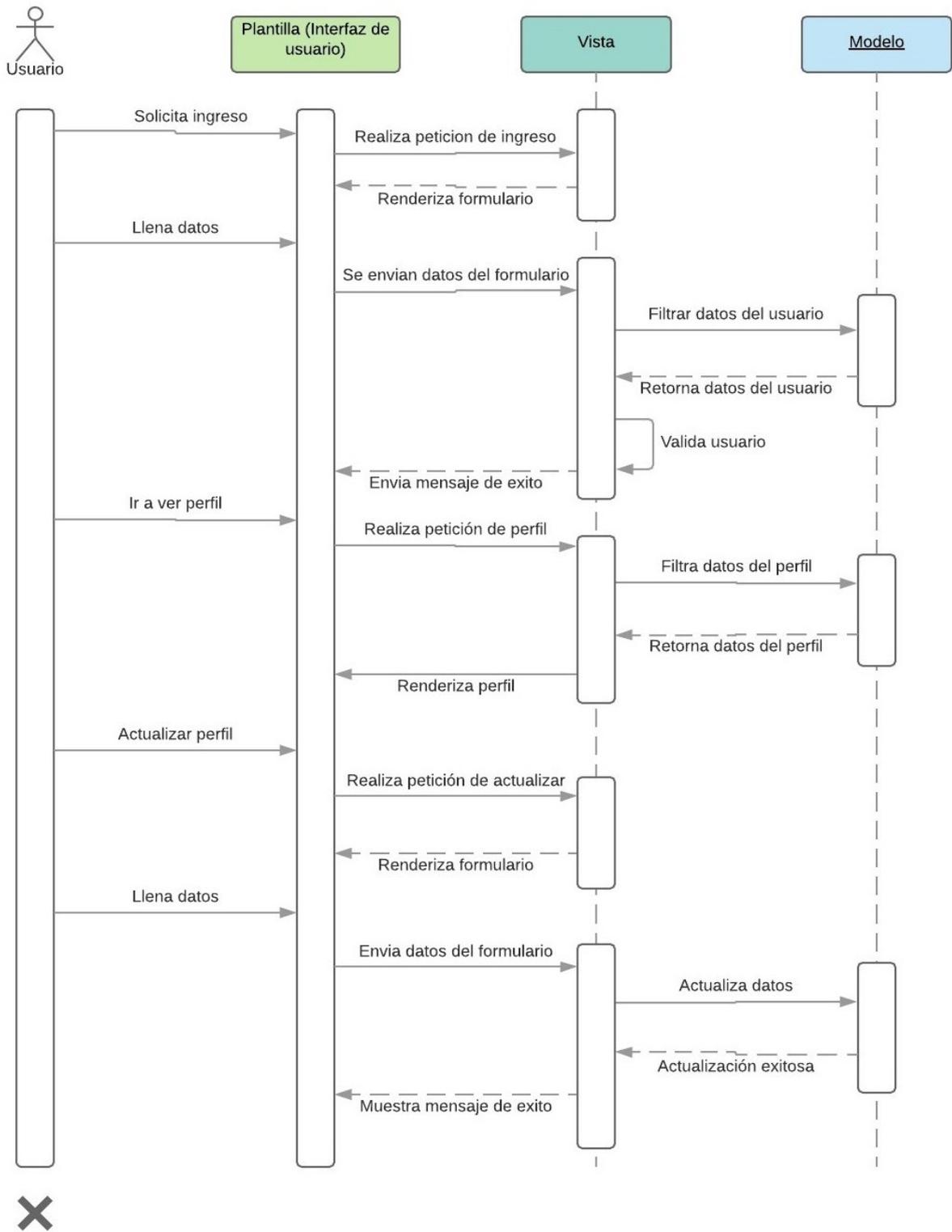


Imagen 7. Diagrama de secuencia del caso de uso 4

7.4.2 Caso de uso 12, Creación de ofertas a propuestas

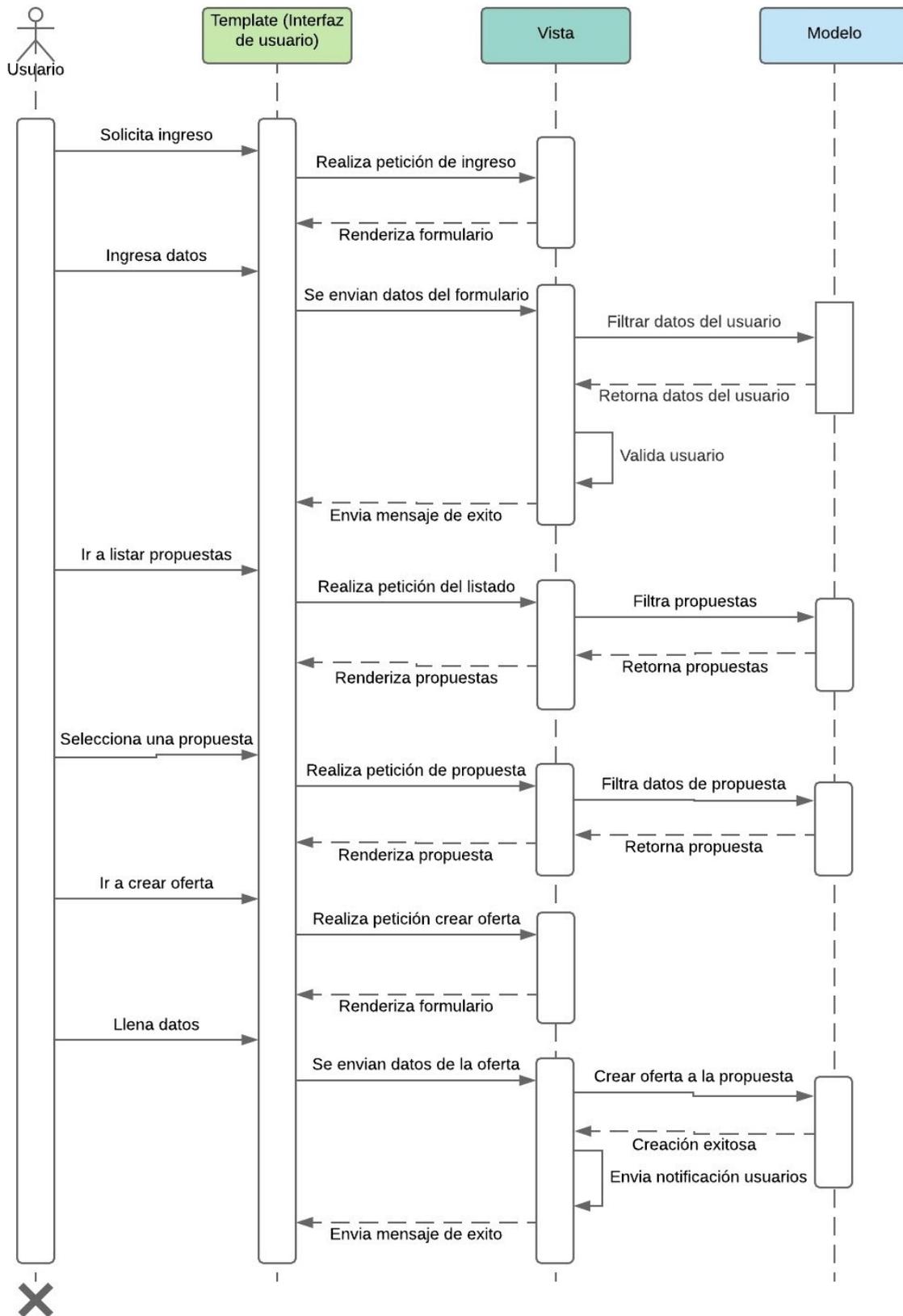


Imagen 8. Diagrama de secuencia del caso de uso 12

7.4.3 Caso de uso 24, Envío de mensajes

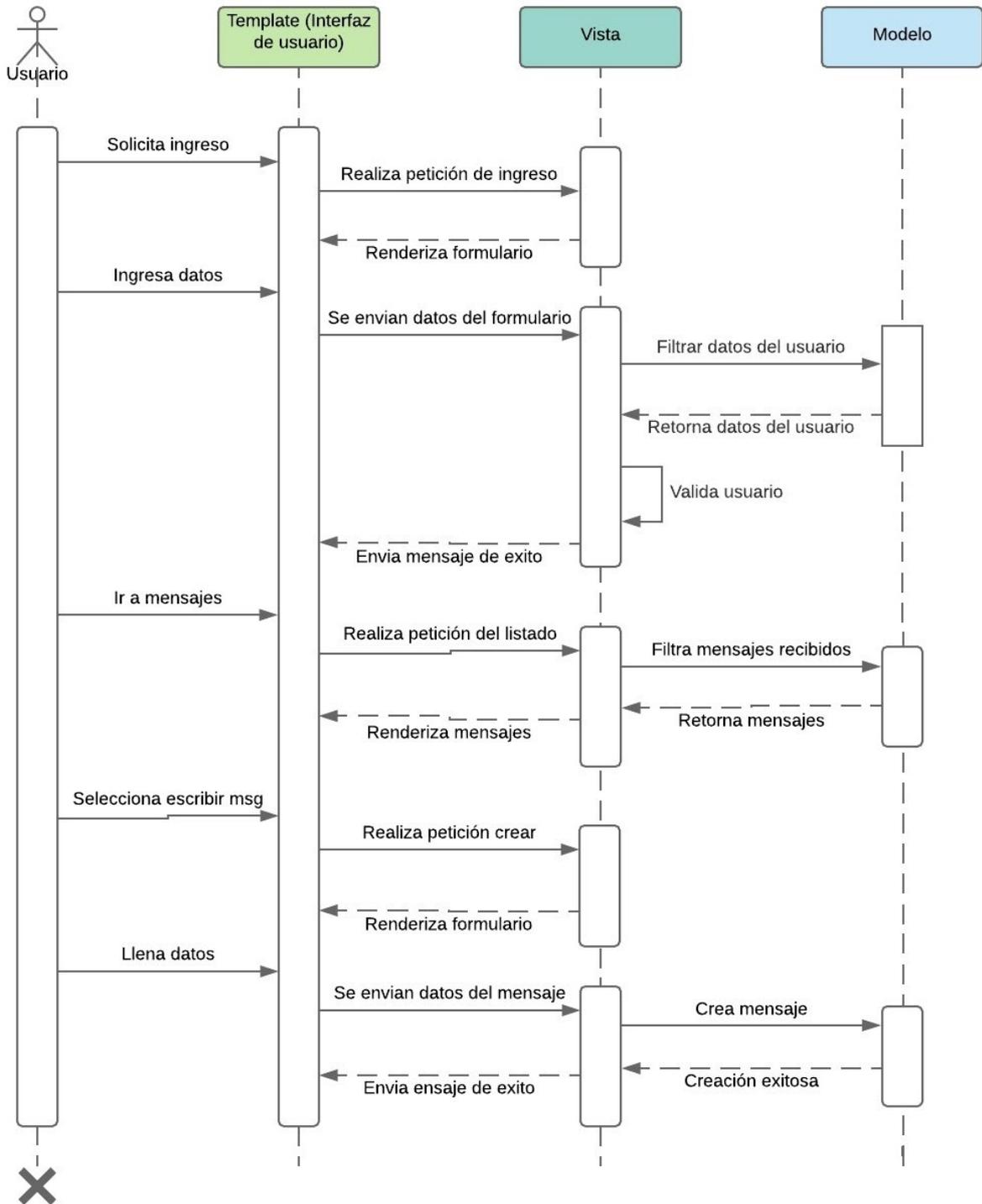


Imagen 9. Diagrama de secuencia del caso de uso 24

7.5. Casos de pruebas

Casos de pruebas						
Módulo	Número	Descripción	Precondiciones	Acciones	Datos ingresados	Resultado esperado
Módulo de usuario	CP-1	Registrar usuario		- El usuario debe ingresar a la sección de registro de usuarios	- Correo electrónico - Nombre de usuario - Contraseña - Repetir contraseña - Aceptar políticas y tratamiento de datos	El usuario es registrado exitosamente
	CP-2	Registrar usuario con correo repetido	- Debe haber registrado un usuario con el mismo correo electrónico de la prueba	- El usuario debe ingresar a la sección de registro de usuarios	- Correo electrónico - Nombre de usuario - Contraseña - Repetir contraseña - Aceptar políticas y tratamiento de datos	Mensaje de error por correo electrónico repetido
	CP-3	Registrar usuario con nombre de usuario repetido	- Debe haber registrado un usuario con el mismo nombre de usuario de la prueba	- El usuario debe ingresar a la sección de registro de usuarios	- Correo electrónico - Nombre de usuario - Contraseña - Repetir contraseña - Aceptar políticas y tratamiento de datos	Mensaje de error por nombre de usuario repetido
	CP-4	Registrar usuario con contraseña de menos de 8 caracteres		- El usuario debe ingresar a la sección de registro de usuarios	- Correo electrónico - Nombre de usuario - Contraseña con menos de 8 caracteres - Repetir contraseña con menos de 8 caracteres - Aceptar políticas y tratamiento de datos	Mensaje de error por contraseña muy corta

CP-5	Registrar usuario con diferente contraseña en el formulario		- El usuario debe ingresar a la sección de registro de usuarios	- Correo electrónico - Nombre de usuario - Contraseña - Repetir contraseña (diferente al campo anterior) - Aceptar políticas y tratamiento de datos	Mensaje de error por contraseña diferente
CP-6	Registrar usuario con sin marcar política de privacidad		- El usuario debe ingresar a la sección de registro de usuarios	- Correo electrónico - Nombre de usuario - Contraseña - Repetir contraseña - No aceptar políticas y tratamiento de datos	Mensaje de validación pidiendo al usuario marcar el campo de aceptar políticas y tratamiento de datos
CP-7	Ingreso de usuario ya registrado	- Debe haber un usuario registrado con los datos ingresados	- El usuario debe ingresar a la sección de ingreso de usuarios	- Correo electrónico o nombre de usuario - Contraseña	El sistema debe darle acceso a su perfil de usuario, ingreso exitoso
CP-8	Ingreso de usuario inexistente		- El usuario debe ingresar a la sección de ingreso de usuarios	- Correo electrónico o nombre de usuario - Contraseña	El sistema debe mostrar un mensaje de que la persona no existe
CP-9	Ingreso de usuario con correo equivocado	- Debe haber un usuario registrado con los datos ingresados	- El usuario debe ingresar a la sección de ingreso de usuarios	- Correo electrónico equivocado - Contraseña	El sistema debe mostrar un mensaje de que el correo no se encuentra registrado
CP-10	Ingreso de usuario con nombre de usuario equivocado	- Debe haber un usuario registrado con los datos ingresados	- El usuario debe ingresar a la sección de ingreso de usuarios	- Nombre de usuario equivocado - Contraseña	El sistema debe mostrar un mensaje de que el nombre de usuario no se encuentra registrado
CP-11	Ingreso de usuario con contraseña diferente	- Debe haber un usuario registrado con los datos ingresados	- El usuario debe ingresar a la sección de ingreso de usuarios	- Correo electrónico o nombre de usuario - Contraseña diferente a la registrada	El sistema debe mostrar un mensaje de que la contraseña no es válida

CP-12	Ver perfil de usuario	- Debe haber un usuario registrado y logueado en la aplicación	- El usuario debe ingresar a su perfil de usuario		Se debe mostrar el perfil de usuario
CP-13	Ver perfil de usuario público	- No debe haber un usuario logueado en la aplicación	- El usuario debe ingresar a un perfil de usuario		Se debe mostrar el perfil de usuario ocultando los datos de información confidencial (número de horas, teléfono, correo electrónico, edad, fecha de nacimiento)
CP-14	Actualizar perfil	- Debe haber un usuario registrado y logueado en el sistema	- El usuario debe ingresar a su perfil de usuario y dar click en editar perfil	<ul style="list-style-type: none"> - Nombres - Apellidos - Foto de perfil - Celular - Descripción - Dirección - Ciudad - Departamento - País - Habilidades 	El sistema debe actualizar el usuario mostrando un mensaje de éxito
CP-15	Actualizar contraseña	- Debe haber un usuario registrado y logueado en el sistema	- El usuario debe ingresar a su perfil de usuario y dar click en editar contraseña	<ul style="list-style-type: none"> - Contraseña actual - Nueva contraseña - Repetir contraseña 	El sistema debe actualizar la contraseña mostrando un mensaje de éxito
CP-16	Actualizar contraseña ingresando en el campo contraseña actual una diferente	- Debe haber un usuario registrado y logueado en el sistema	- El usuario debe ingresar a su perfil de usuario y dar click en editar contraseña	<ul style="list-style-type: none"> - Contraseña actual diferente a la real - Nueva contraseña - Repetir contraseña 	El sistema debe mostrar un mensaje que no pudo actualizar la contraseña porque no coincidía con la actual
CP-17	Actualizar contraseña ingresando en el campo de repetir	- Debe haber un usuario registrado y logueado en el sistema	- El usuario debe ingresar a su perfil de usuario y dar click en editar contraseña	<ul style="list-style-type: none"> - Contraseña actual - Nueva contraseña - Repetir contraseña diferente a la nueva 	El sistema debe mostrar un mensaje que no pudo actualizar la contraseña porque no coincidía la

		contraseña una diferente a la nueva				nueva contraseña
Módulo de Intercambios	CP-18	Listar propuestas de usuarios	- Debe haber un usuario registrado y logueado en el sistema	- El usuario debe ingresar a listar sus propuestas		El sistema debe mostrar el listado de propuestas creadas por el usuario
	CP-19	Crear propuestas	- Debe haber un usuario registrado y logueado en el sistema	- El usuario debe ingresar a listar sus propuestas y dar click en crear propuesta	- Título - Descripción corta - Descripción larga - Horas - Ciudad - Lugar - Habilidades	El sistema debe crear una propuesta al usuario mostrando un mensaje de éxito
	CP-20	Crear propuestas sin horas disponibles	- Debe haber un usuario registrado y logueado en el sistema	- El usuario debe ingresar a listar sus propuestas y dar click en crear propuesta	- Título - Descripción corta - Descripción larga - Horas (sin horas disponibles) - Ciudad - Lugar - Habilidades	El sistema debe mostrar un mensaje de error de que el usuario no posee horas para crear la propuesta
	CP-21	Actualizar propuestas	- Debe haber un usuario registrado y logueado en el sistema - Debe haber una propuesta del usuario logueado	- El usuario debe ingresar a listar sus propuestas y dar click en editar una propuesta	- Título - Descripción corta - Descripción larga - Horas - Ciudad - Lugar - Habilidades	El sistema debe editar la propuesta mostrando un mensaje de edición exitosa
	CP-22	Actualizar propuestas aceptadas	- Debe haber un usuario registrado y logueado en el sistema - Debe haber una propuesta del usuario	- El usuario debe ingresar a listar sus propuestas y dar click en editar una propuesta	- Título - Descripción corta - Descripción larga - Horas - Ciudad - Lugar	El sistema debe mostrar un mensaje de que no se puede actualizar la propuesta aceptada

		logueado - La propuesta debe tener una oferta aceptada		- Habilidades	
CP-23	Buscador de propuestas	- Debe haber un usuario registrado y logueado en la aplicación - Debe haber propuestas de otros usuarios creadas en la aplicación	- El usuario debe ingresar a la sección de buscar propuestas de otros usuarios	- Departamento - Ciudad - Lugar - Habilidades	El sistema debe mostrar las propuestas de acuerdo a los criterios de búsqueda seleccionados
CP-24	Ver propuesta individual	- Debe haber un usuario registrado y logueado en la aplicación - Debe haber propuestas creadas	- El usuario debe ingresar a listar sus propuestas y dar click en ver propuesta, o debe ingresar en la sección de buscar propuestas y dar click en ver propuesta		El sistema debe mostrar la propuesta como página con información de la misma y del usuario más detallada
CP-25	Ver oferta	- Debe haber un usuario registrado y logueado en la aplicación - Debe haber propuestas creadas - Debe haber una oferta creada por el usuario	- El usuario debe ingresar a listar sus propuestas y dar click en ver propuesta, o debe ingresar en la sección de buscar propuestas y dar click en ver propuesta		El sistema debe mostrar debajo de la propuesta, la oferta.
CP-26	Hacer oferta a propuesta	- Debe haber un usuario registrado y logueado en la aplicación - Debe haber propuestas de otros usuarios creadas en la aplicación	- El usuario debe ingresar a la sección de buscar propuestas y dar click en ver propuesta - El usuario debe dar click en hacer oferta	- Horas - Descripción	El sistema debe crear la oferta a la propuesta mostrando un mensaje de propuesta creada exitosamente

CP-27	Hacer oferta a propuesta sin horas disponibles	<ul style="list-style-type: none"> - Debe haber un usuario registrado y logueado en la aplicación - Debe haber propuestas de otros usuarios creadas en la aplicación 	<ul style="list-style-type: none"> - El usuario debe ingresar a la sección de buscar propuestas y dar click en ver propuesta - El usuario debe dar click en hacer oferta 	<ul style="list-style-type: none"> - Horas (no disponibles) - Descripción 	El sistema debe mostrar un mensaje de que no se pudo crear la oferta ya que no hay horas disponibles
CP-28	Hacer oferta a propuesta con una oferta ya aceptada	<ul style="list-style-type: none"> - Debe haber un usuario registrado y logueado en la aplicación - Debe haber propuestas de otros usuarios creadas en la aplicación - La propuesta debe tener una oferta aceptada 	<ul style="list-style-type: none"> - El usuario debe ingresar a la sección de buscar propuestas y dar click en ver propuesta - El usuario debe dar click en hacer oferta 	<ul style="list-style-type: none"> - Horas - Descripción 	El sistema debe mostrar un mensaje de que no se pudo crear la oferta ya que ya tiene una oferta aceptada
CP-29	Actualizar oferta a propuesta	<ul style="list-style-type: none"> - Debe haber un usuario registrado y logueado en la aplicación - Debe haber propuestas de otros usuarios creadas en la aplicación - El usuario logueado debe tener una oferta creada en la propuesta 	<ul style="list-style-type: none"> - El usuario debe ingresar a la sección de buscar propuestas y dar click en ver propuesta - El usuario debe dar click en editar oferta 	<ul style="list-style-type: none"> - Horas - Descripción 	El sistema debe actualizar la oferta mostrando un mensaje de éxito

CP-30	Actualizar oferta a propuesta finalizada	<ul style="list-style-type: none"> - Debe haber un usuario registrado y logueado en la aplicación - Debe haber propuestas de otros usuarios creadas en la aplicación - El usuario logueado debe tener una oferta creada en la propuesta 	<ul style="list-style-type: none"> - El usuario debe ingresar a la sección de buscar propuestas y dar click en ver propuesta - El usuario debe dar click en editar oferta 	<ul style="list-style-type: none"> - Horas - Descripción 	El sistema debe mostrar un mensaje de que no se puede editar la oferta
CP-31	Eliminar oferta a propuesta	<ul style="list-style-type: none"> - Debe haber un usuario registrado y logueado en la aplicación - Debe haber propuestas de otros usuarios creadas en la aplicación - El usuario logueado debe tener una oferta creada en la propuesta 	<ul style="list-style-type: none"> - El usuario debe ingresar a la sección de buscar propuestas y dar click en ver propuesta - El usuario debe dar click en eliminar oferta 		El sistema debe eliminar la oferta mostrando un mensaje de éxito
CP-32	Eliminar oferta a propuesta finalizada	<ul style="list-style-type: none"> - Debe haber un usuario registrado y logueado en la aplicación - Debe haber propuestas de otros usuarios creadas en la aplicación - El usuario logueado debe tener una oferta creada en la propuesta 	<ul style="list-style-type: none"> - El usuario debe ingresar a la sección de buscar propuestas y dar click en ver propuesta - El usuario debe dar click en eliminar oferta 		El sistema debe mostrar un mensaje de que no se puede eliminar la oferta

	CP-33	Aceptar oferta	<ul style="list-style-type: none"> - Debe haber un usuario registrado y logueado en la aplicación - Debe haber una propuesta creada del usuario logueado - Debe haber una oferta de otro usuario en la propuesta 	<ul style="list-style-type: none"> - El usuario debe ingresar en sus propuestas y hacer click en ver propuesta - El usuario debe dar click en aceptar oferta 		El sistema debe mostrar un mensaje de que la oferta fue aceptada
	CP-34	Rechazar oferta	<ul style="list-style-type: none"> - Debe haber un usuario registrado y logueado en la aplicación - Debe haber una propuesta creada del usuario logueado - Debe haber una oferta de otro usuario en la propuesta 	<ul style="list-style-type: none"> - El usuario debe ingresar en sus propuestas y hacer click en ver propuesta - El usuario debe dar click en rechazar oferta 		El sistema debe mostrar un mensaje de que la oferta fue rechazada
	CP-35	Finalizar oferta	<ul style="list-style-type: none"> - Debe haber un usuario registrado y logueado en la aplicación - Debe haber una propuesta creada del usuario logueado - Debe haber una oferta aceptada de otro usuario en la propuesta 	<ul style="list-style-type: none"> - El usuario debe ingresar en sus propuestas y hacer click en ver propuesta - El usuario debe dar click en finalizar propuesta 		El sistema debe mostrar un mensaje de que la propuesta ha sido finalizada transfiriendo las horas al usuario de la oferta
Módulo de mensajes	CP-36	Listar mensajes enviados de un usuario	<ul style="list-style-type: none"> - Debe haber un usuario registrado y logueado en el sistema 	<ul style="list-style-type: none"> - El usuario debe ingresar a listar sus mensajes enviados 		El sistema debe mostrar el listado de mensajes enviados

CP-37	Ver mensaje	<ul style="list-style-type: none"> - Debe haber un usuario registrado y logueado en el sistema - El usuario debe haber enviado al menos un mensaje 	<ul style="list-style-type: none"> - El usuario debe ingresar a listar sus mensajes enviados - El usuario debe ingresar en un mensaje 		El sistema debe mostrar el mensaje en detalle
CP-38	Responder mensaje	<ul style="list-style-type: none"> - Debe haber un usuario registrado y logueado en el sistema - El usuario debe haber enviado al menos un mensaje 	<ul style="list-style-type: none"> - El usuario debe ingresar a listar sus mensajes enviados o recibidos - El usuario debe ingresar en un mensaje - El usuario debe responder el mensaje 		El sistema debe responder el mensaje mostrando un mensaje de éxito
CP-39	Eliminar mensajes	<ul style="list-style-type: none"> - Debe haber un usuario registrado y logueado en el sistema - El usuario debe haber enviado o recibido al menos un mensaje 	<ul style="list-style-type: none"> - El usuario debe ingresar a listar sus mensajes enviados o recibidos - El usuario debe seleccionar el mensaje que desea eliminar y dar click en eliminar 		El sistema debe marcar el mensaje como eliminado, mostrando un mensaje de éxito
CP-40	Listar mensajes eliminados de un usuario	<ul style="list-style-type: none"> - Debe haber un usuario registrado y logueado en el sistema 	<ul style="list-style-type: none"> - El usuario debe ingresar a listar sus mensajes eliminados 		El sistema debe mostrar el listado de mensajes eliminados
CP-41	Recuperar mensajes eliminados de un usuario	<ul style="list-style-type: none"> - Debe haber un usuario registrado y logueado en el sistema - Debe tener al menos un mensaje eliminado 	<ul style="list-style-type: none"> - El usuario debe ingresar a listar sus mensajes eliminados - El usuario debe seleccionar el mensaje que desea recuperar 		El sistema debe recuperar el mensaje mostrando un mensaje de éxito
CP-42	Listar mensajes recibidos de un	<ul style="list-style-type: none"> - Debe haber un usuario registrado y 	<ul style="list-style-type: none"> - El usuario debe ingresar a listar sus mensajes recibidos 		El sistema debe mostrar el listado de mensajes

		usuario	logueado en el sistema			recibidos
	CP-43	Escribir un mensaje	- Debe haber un usuario registrado y logueado en el sistema	- El usuario debe ingresar a la sección de escribir un mensaje	- Usuario - Asunto - Mensaje	El sistema debe enviar el mensaje mostrando un mensaje de éxito
Módulo de calificaciones	CP-44	Ver calificaciones pendientes (usuario que creó propuesta)	- Debe haber un usuario registrado y logueado en el sistema - Debe haber calificaciones pendientes por realizar (creación de propuesta)	- El usuario debe ingresar a listar las calificaciones pendientes		El sistema debe mostrar el listado de propuestas pendientes por calificar
	CP-45	Ver calificaciones pendientes (usuario que hizo oferta)	- Debe haber un usuario registrado y logueado en el sistema - Debe haber calificaciones pendientes por realizar (creación de oferta)	- El usuario debe ingresar a listar las calificaciones pendientes		El sistema debe mostrar el listado de propuestas pendientes por calificar
	CP-46	Ver calificaciones realizadas	- Debe haber un usuario registrado y logueado en el sistema - Debe haber calificaciones realizadas	- El usuario debe ingresar a listar las calificaciones realizadas		El sistema debe mostrar el listado de propuestas realizadas

	CP-47	Calificar usuario	<ul style="list-style-type: none"> - Debe haber un usuario registrado y logueado en el sistema - Debe haber calificaciones pendientes por realizar (propuestas finalizadas) 	<ul style="list-style-type: none"> - El usuario debe ingresar a listar las calificaciones pendientes - El usuario debe ingresar a calificar un usuario 	<ul style="list-style-type: none"> - Calificación de 1 a 5 - Mensaje 	La calificación debe realizarse mostrando un mensaje de éxito
--	-------	-------------------	---	--	--	---

Tabla 5. Casos de pruebas

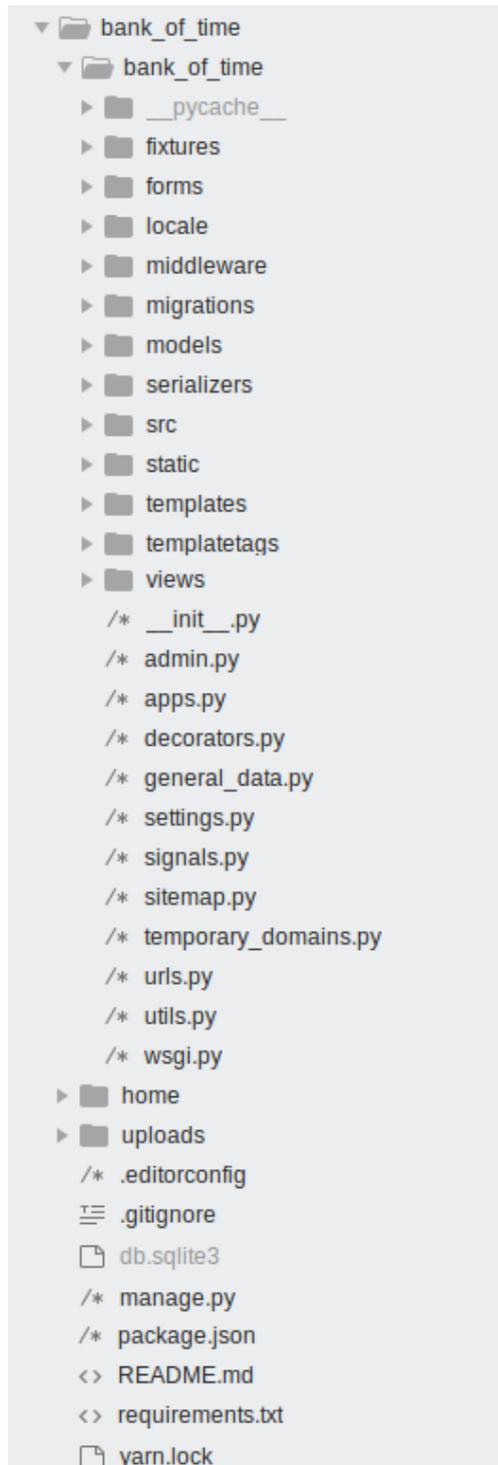
7.6. Registro de defectos

Nro	Descripción	Fase de inyección	Fase de remoción	Tiempo en solucionar (horas)
2	Diagramación casos de uso (mensajes)	diseño	revisión diseño	0:32:00
1	tipos de flechas casos de uso	diseño	revisión diseño	0:12:00
1	copys registro o ingreso caso de uso	diseño	revisión diseño	0:01:00
1	soluciona transferencia horas extiende de ver ofertas	diseño	revisión diseño	0:02:00
1	Caso de uso que extiende de eliminar mensaje	diseño	revisión diseño	0:01:00
1	Cambio textos de precondición en casos de uso	diseño	revisión diseño	0:01:00
1	Elimina curso alternativo que no aplica en caso de uso 27	diseño	revisión diseño	0:01:00
12			Tiempo total	00:50:00

Tabla 6. Defectos en Diseño

8. Codificación

La codificación del proyecto se realizó de acuerdo a la planeación, diseño, calendario del proyecto y los estándares, guías y formatos proveídos por PSP.



8.1. Estructura usada en la codificación

- **Modelos:** En la carpeta models.
- **Lógica:** Las vistas están ubicadas en la carpeta views, también hay lógica de programación en la carpeta templatetags y en los archivos signals.py y general_data.py
- **Plantillas y Frontend:** Las plantillas HTML están ubicadas en la carpeta templates, el Frontend en realizado en Reactjs está ubicado en la carpeta src.
- **Estilos:** Los estilos en cascada (CSS) están ubicados en la carpeta src.
- **Imágenes:** Las imágenes subidas al sistema están ubicadas en la carpeta uploads.
- **Ruteo:** Las urls o ruteo que lleva a las vistas estan ubicadas en el archivo urls.py
- **Permisos:** Los permisos de usuario se validan en el archivo decorators.py.
- **Entorno:** Las dependencias requeridas para el entorno de Python (Django), Reactjs y estilos (CSS) están en los archivos requirements.txt y en package.json.
- **Otros archivos:** El framework Django genera algunos archivos y carpetas de configuración por defecto.

Imagen 10. Estructura de archivos

8.2. Tamaño de la aplicación

El tamaño de la aplicación se definió en líneas de código (LOC), PSP hace referencia a que el tamaño de la aplicación se debe hacer tomando en cuenta los lenguajes de programación utilizados, dado que HTML5 y CSS3 no son lenguajes, se decidió separar el tamaño de la aplicación en 2.

El tamaño de la aplicación en Líneas de código para los lenguajes de programación usados (Python y Javascript - Reactjs) fue de: **5.667**

Si sumamos la estructura en HTML y estilos (CSS), el tamaño en Líneas de código fue de: **8.180**

8.3. Imágenes del funcionamiento de la Aplicación

8.3.1. Módulo de usuarios



The image shows a login form with the following elements:

- Title:** Iniciar sesión
- Text:** Por favor inicia sesión con una cuenta de otra red social. O, [regístrate](#) como usuario de Banco del Tiempo e inicia sesión a continuación:
- Username field:** Labeled "Iniciar sesión :", containing the text "valerock".
- Password field:** Labeled "Contraseña :", containing seven dots.
- Remember me:** A checkbox labeled "Recordarme :".
- Link:** A link labeled "¿OLVIDASTE TU CONTRASEÑA?".
- Submit button:** A blue button labeled "INICIAR SESIÓN" with a right-pointing arrow.

Imagen 11. Formulario de inicio de sesión

Regístrate

¿Ya tienes una cuenta? Por favor [inicia sesión](#).

Correo electrónico :
Correo electrónico

Nombre de usuario : Fecha de nacimiento
(YYYY/MM/DD):
valerock Fecha de nacimiento (YYYY/MM)

Contraseña : Contraseña (de nuevo) :
..... Contraseña (de nuevo)

Acepto los [Términos y condiciones](#) y la [Política de privacidad](#)

REGÍSTRATE >

Imagen 12. Formulario de registro de usuario

Perfil de usuario



Mi tiempo: 180

ACTUALIZAR PASSWORD

ACTUALIZAR PERFIL

Nombre	Jhon Valero
Correo electrónico	micorreo@gmail.com
Nombre de usuario	valerock
Fecha de nacimiento	1990/01/26
Celular	333 333 3333
Ciudad	Dosquebradas, Risaralda, Colombia
Dirección	

Tu Tiempo, Celular y Dirección estan ocultas para otros usuarios.

Mis habilidades

Matematicas Ingles Sistemas

Sobre mi

Ingeniero de sistemas

Imagen 13. Perfil de un usuario

Actualizar perfil

Complete su perfil llenando los siguientes campos, incluyendo una imagen y una habilidad como mínimo.

Información personal

IMAGEN DE PERFIL /uploads/crop.jpeg

Nombre Jhon	Apellidos Valero	Fecha de nacimiento (YYYY/MM/DD) 1990/01/01
Nombre de usuario valerock	Correo electrónico micorreo@gmail.com	Celular 3333333333

Descripción

Estoy ubicado en ...

Pais Colombia	Departamento/Estado Risaralda	Ciudad Dosquebradas
-------------------------	---	-------------------------------

Dirección

Mis habilidades

Matematicas x
Inglés x
Sistemas x
Añadir habilidad

ACTUALIZAR PERFIL

Imagen 14. Formulario de actualizar datos de usuario

← REGRESAR

Cambiar Contraseña

Contraseña actual :

Contraseña actual

Nueva contraseña :

Nueva contraseña

Nueva contraseña (de nuevo) :

Nueva contraseña (de nuevo)

CAMBIAR CONTRASEÑA

Imagen 15. Formulario de actualizar contraseña

8.3.2 Módulo de intercambios

Mis Publicaciones

[VER ACTIVAS](#) [VER TERMINADAS](#) [+ CREAR PUBLICACION](#)

Solicito Clases de Musica  

08/02/2020 - Dosquebradas, Risaralda

Solicito persona que me de clases de Matematicas basicas

120 

[VER MAS](#)

Solicito Clases de Pintura  

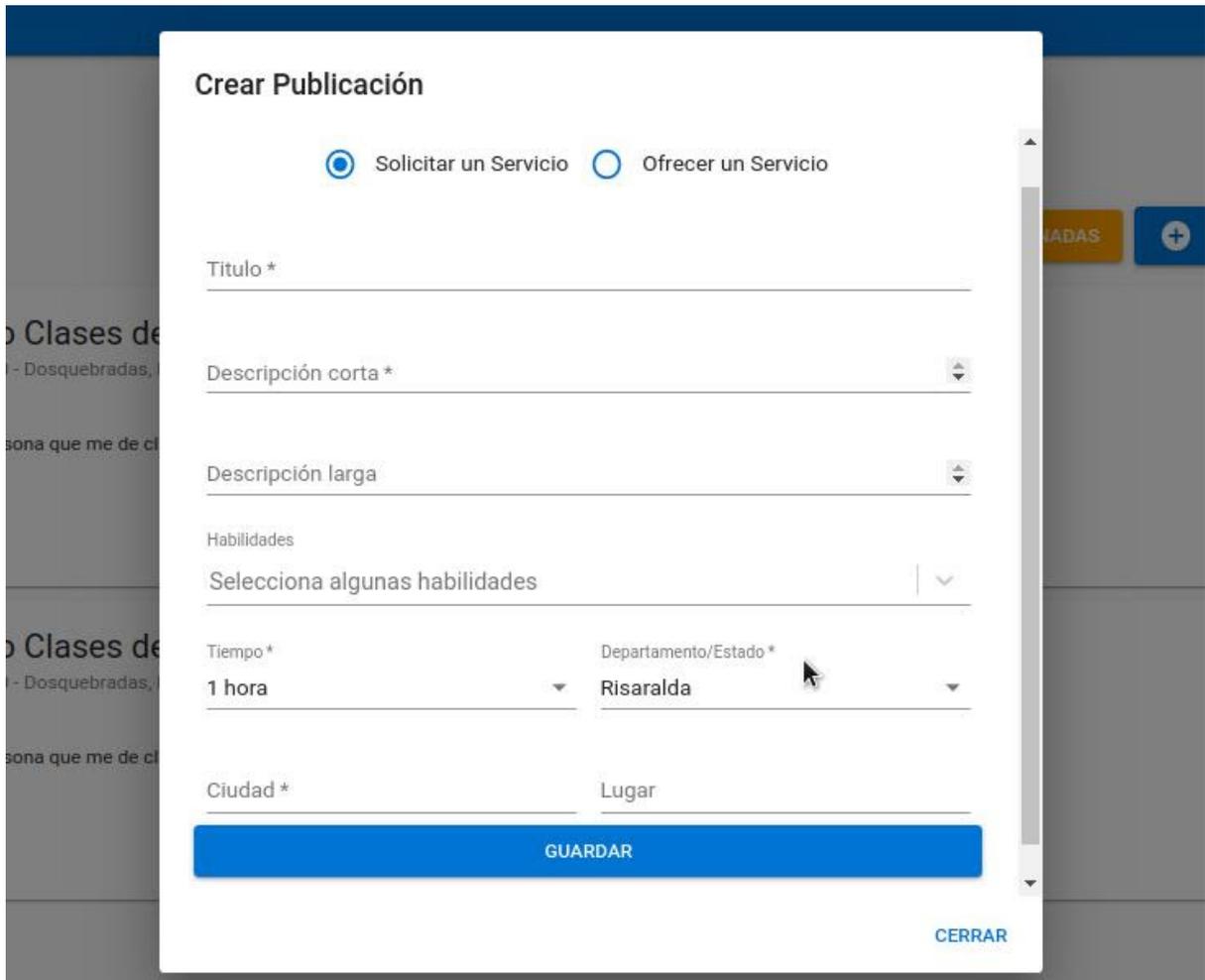
08/02/2020 - Dosquebradas, Risaralda

Solicito persona que me de clases de pintura al oleo

120 

[VER MAS](#)

Imagen 16. Lista de propuestas



Crear Publicación

Solicitar un Servicio Ofrecer un Servicio

Título *

Descripción corta *

Descripción larga

Habilidades

Selecciona algunas habilidades

Tiempo * Departamento/Estado *

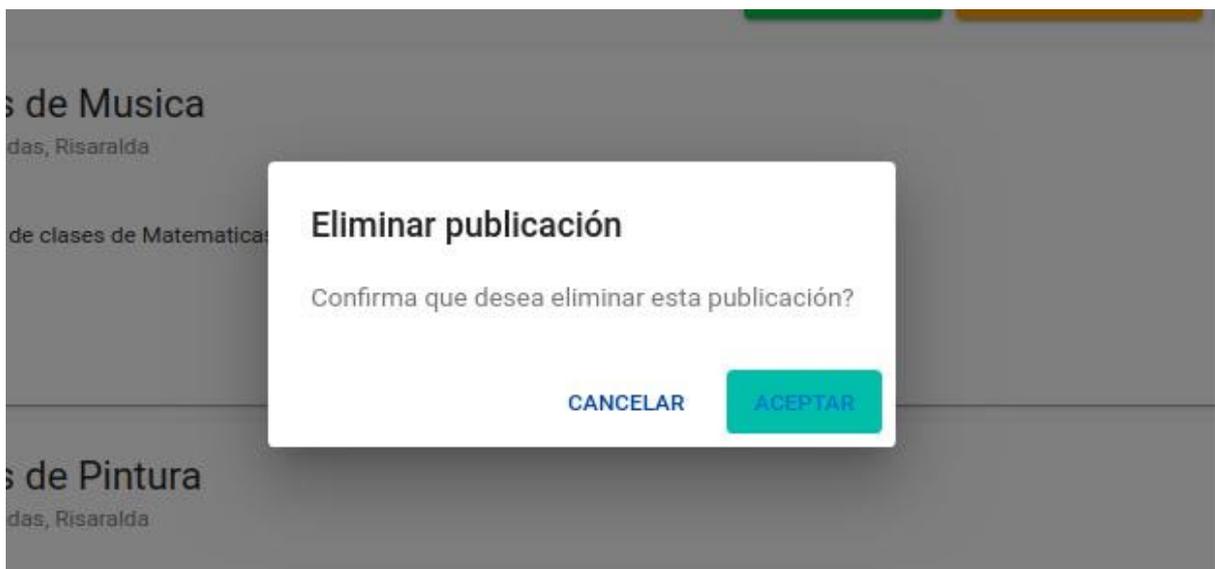
1 hora Risaralda

Ciudad * Lugar

GUARDAR

CERRAR

Imagen 17. Formulario de crear propuestas



Eliminar publicación

Confirma que desea eliminar esta publicación?

CANCELAR ACEPTAR

Imagen 18. Eliminar de propuestas

Actualizar Publicación

Deseas Solicitar u Ofrecer un servicio

Solicitar un Servicio
 Ofrecer un Servicio

Título *

Solicito Clases de Matematicas

Descripción corta *

Solicito Clases de Matematicas

Descripción larga

Habilidades

Matematicas x | v

Tiempo * Departamento/Estado *

2 horas Risaralda

Ciudad * Lugar

Dosquebradas

GUARDAR

[CERRAR](#)

Imagen 19. Formulario de actualizar propuestas

Buscar publicaciones

Tipo de publicación

Solicitudes
 Ofertas

Buscar

Filtrar por Departamento/Estado

Risaralda v

Filtrar por Ciudad

Tiempo Mínimo Tiempo Máximo

Habilidades

Filtrar por habilidades v

LIMPIAR

Jardineria

07/01/2020 - Dosquebradas, Risaralda

Requiero de ayuda basice de jardineria para mi hogar

120
VER MAS

Asesoría en proyectos de software

07/01/2020 - Dosquebradas, Risaralda

Ofrezco asesoria en proyectos de desarrollo de Software web, aplicaciones moviles

120
VER MAS

Imagen 20. Buscador de propuestas

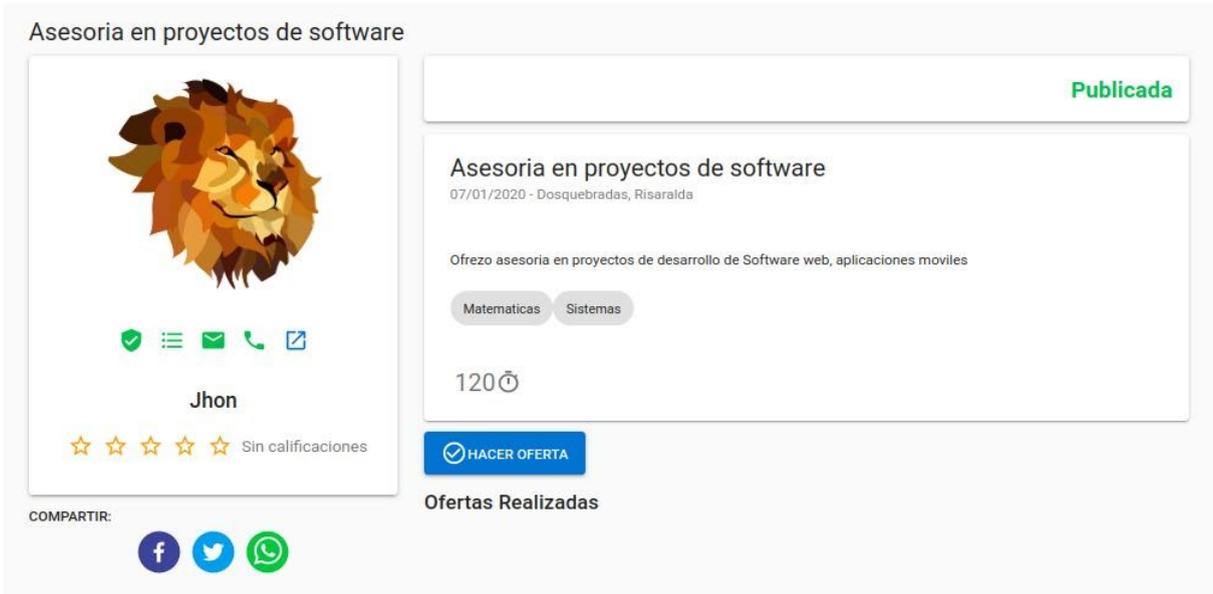


Imagen 21. Vista individual de propuestas

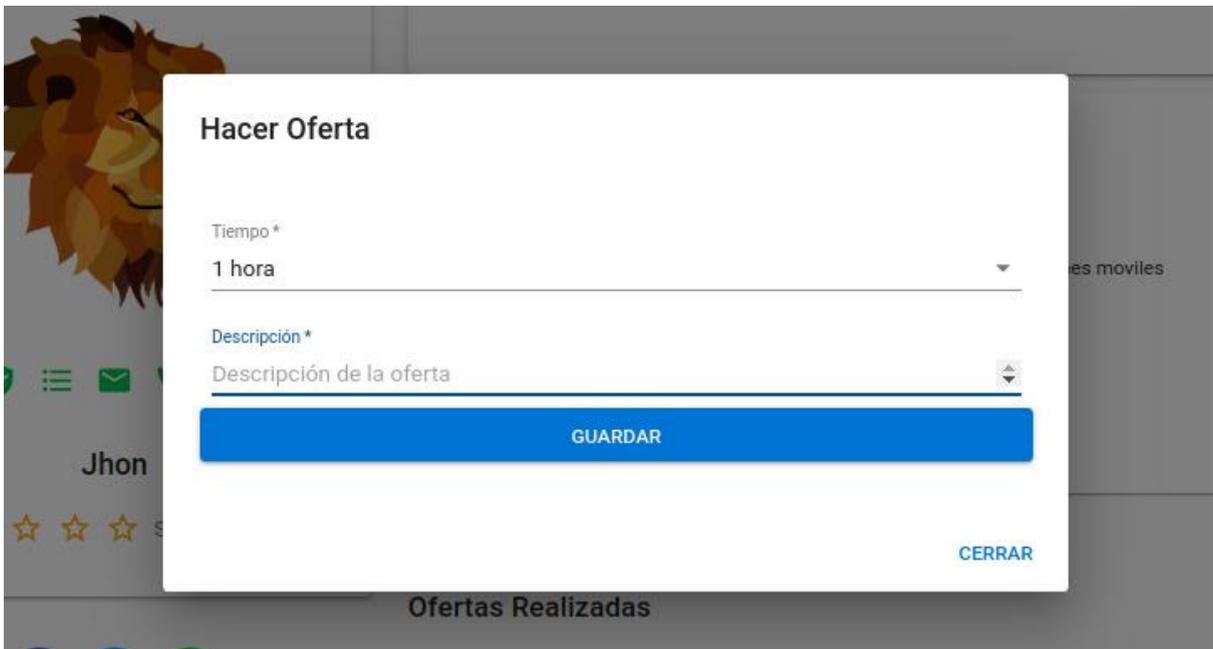


Imagen 22. Formulario de hacer ofertas a propuestas

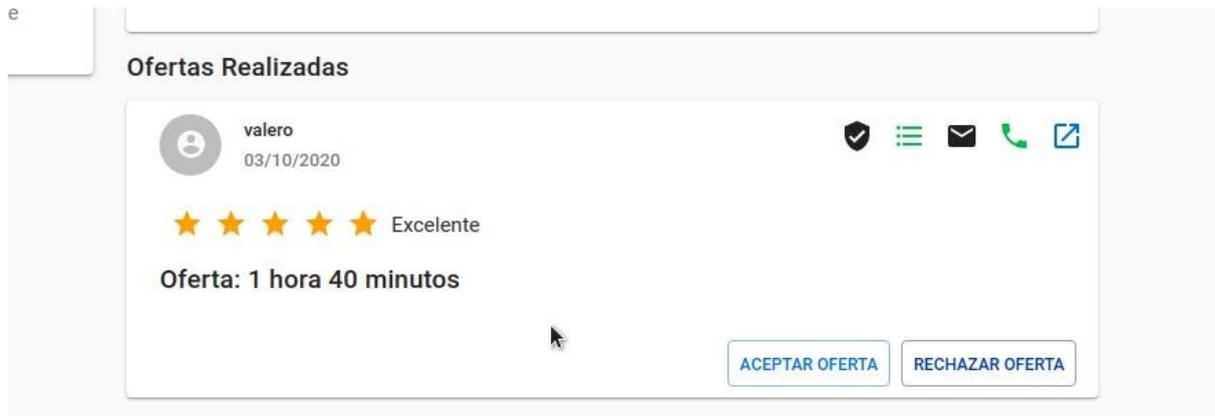


Imagen 23. Vista individual de ofertas

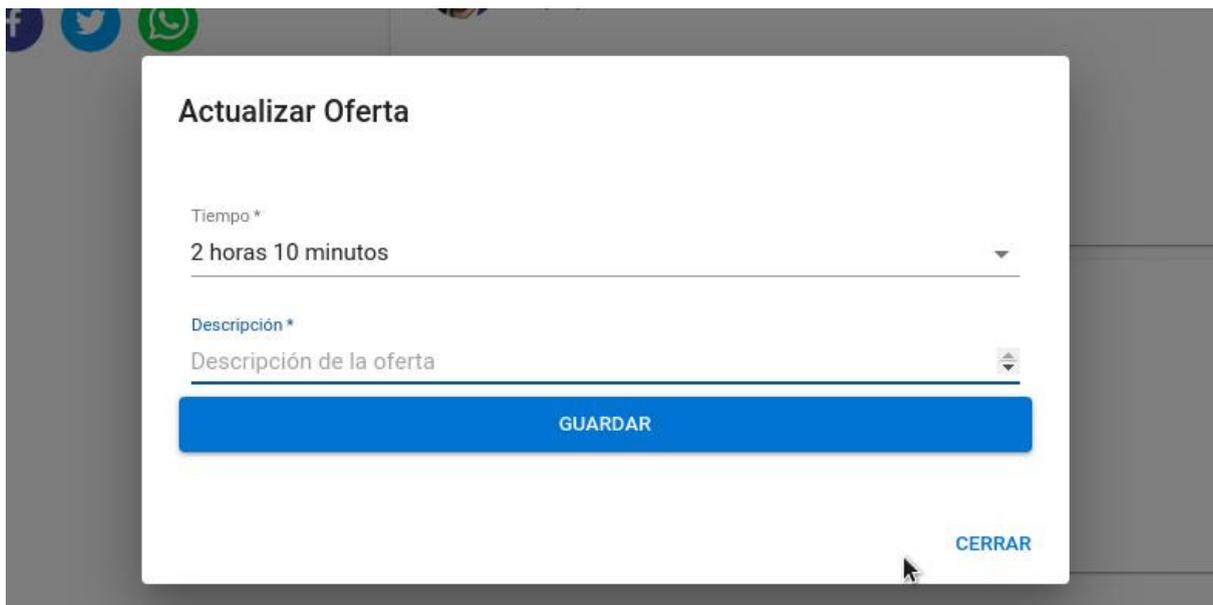


Imagen 24. Formulario de actualizar ofertas

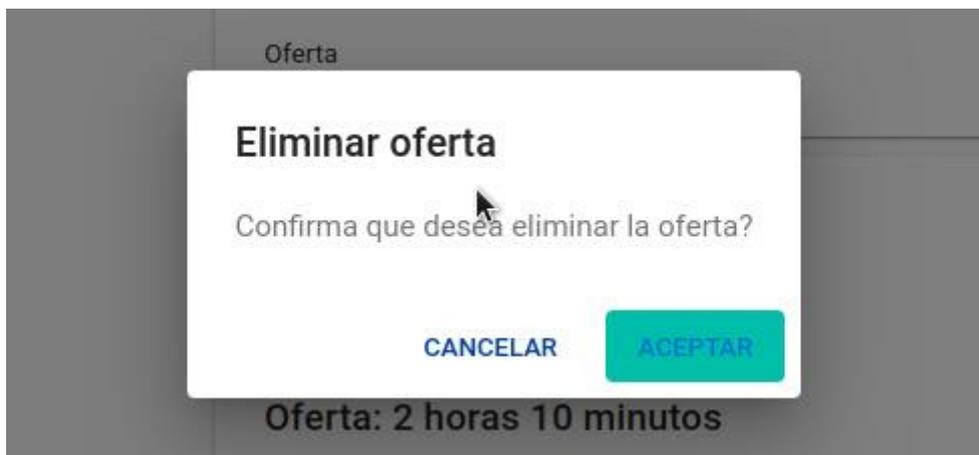


Imagen 25. Eliminar ofertas



Imagen 26. Aceptar ofertas

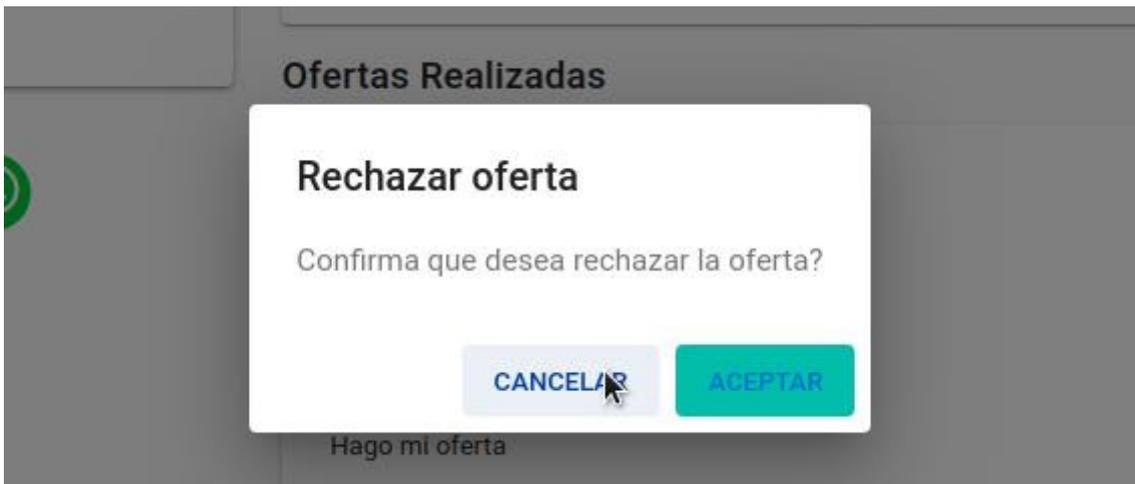


Imagen 27. Aceptar ofertas

8.3.3 Módulo de Mensajes

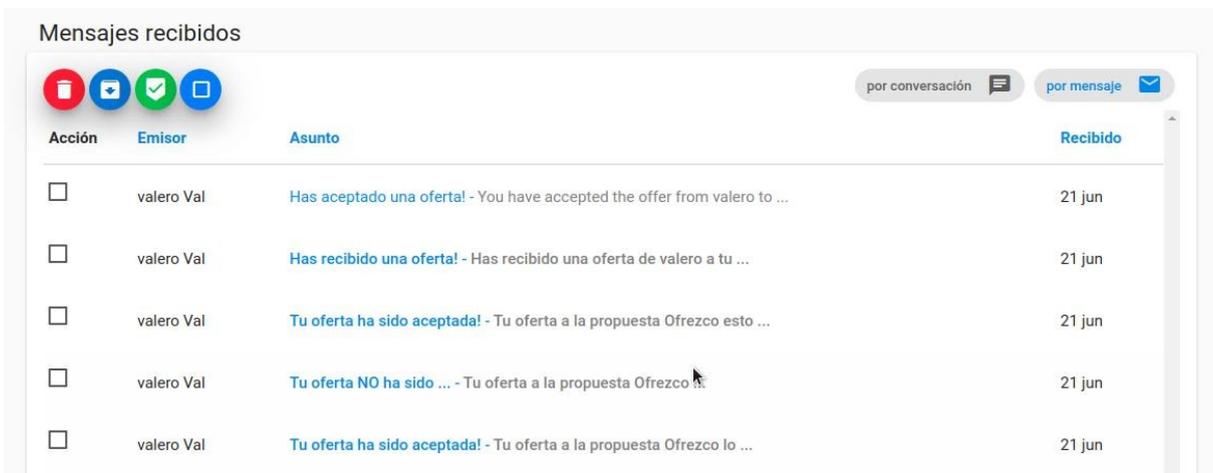


Imagen 28. Lista de mensajes recibidos



Imagen 29. Vista individual de un mensaje



Imagen 30. Formulario para responder mensajes

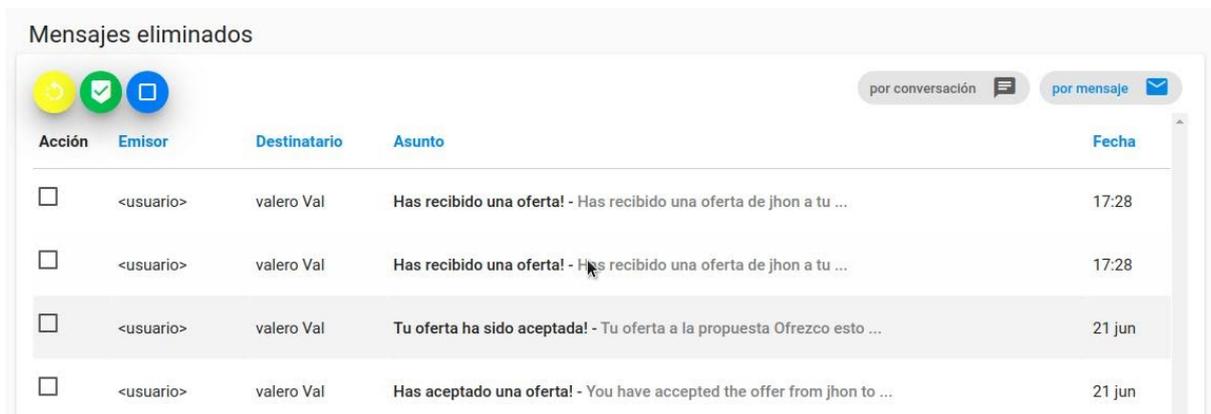


Imagen 31. Lista de mensajes eliminados

Escribe

recipients
 Seleccione un usuario

asunto

contenido

ENVIAR >

Imagen 32. Formulario de enviar mensajes

Mensajes enviados

por conversación por mensaje

Acción	Destinatario	Asunto	Enviado
<input type="checkbox"/>	valero Val	Solicitud de propuesta - propuesta	17:36
<input type="checkbox"/>	<usuario>	Banco del tiempo tiene un ... - Banco del tiempo te regala 180 ...	20 mar

Imagen 33. Lista de mensajes enviados

8.3.3 Módulo de Calificaciones

Calificaciones pendientes

valero

Califica al usuario de la publicación: Ofrezco clases de matematicas

CALIFICAR

valero

Califica al usuario de la publicación: Necesito ayuda en tareas domesticas

CALIFICAR

valero

Califica al usuario de la oferta de tu publicación Ofrezco clases de Pintura

CALIFICAR

Imagen 34. Lista de calificaciones pendientes

iones pendientes

valero

usuario de la

co perro

valero

usuario de la

co esto

valero

valero

Califica a valero

Califica de 1 a 5, siendo 1 la calificación mas baja y 5 la mas alta. La calificación no podrá ser editada.

Calificación *

5

Descripción *

CALIFICAR

CERRAR

Imagen 35. Formulario de calificaciones

Calificaciones realizadas

<p>Reinel</p> <p>★★★★★</p> <p>Recomiendo a esta personal, muy buena su ayuda</p>	<p>German</p> <p>★★★★☆</p> <p>100% recomendado</p>
<p>Sara</p> <p>★★★★★</p>	

Imagen 36. Lista de calificaciones realizadas

9. Compilación y Pruebas

Durante la fase de compilación se solucionaron las fallas de asignación y sintaxis que afectaban la ejecución del programa, en total se corrigieron 89 errores de asignación y 36 de sintaxis, los cuales se contabilizaron en las tablas de log de defectos y de tiempos que se mostraran mejor en la sección de PostMortem.

Durante la fase de pruebas se usaron los casos de prueba definidos en la etapa de diseño (ver tabla), si el resultado obtenido al hacer pruebas en el software no era el esperado se hacía la respectiva corrección, en total se corrigieron 75 defectos en la fase de pruebas, los defectos de la fase de pruebas también se contabilizaron en las tablas de log de defectos y de tiempos.

VII. PostMortem

PSP tiene también un guión para la fase de PostMortem:

Guión del PostMortem (PSP0)

Propósito	Para guiar el proceso postmortem del PSP
Criterios de Entrada	<ul style="list-style-type: none"> – Descripción del problema y declaración de requerimientos – Formulario Resumen del Plan del Proyecto con información del tamaño del programa y tiempo de desarrollo – Log de Registro de Defectos y Tiempos completados – Un programa probado y ejecutado de acuerdo a los Estándares de Codificación y conteo de tamaño

Paso	Actividades	Descripción
1	Registrar defectos	<ul style="list-style-type: none"> – Revisar el Resumen del Plan del Proyecto para verificar que todos los defectos encontrados en cada fase fueron registrados. – Registrar algunos defectos omitidos
2	Consistencia de información de defectos	<ul style="list-style-type: none"> – Revisar que la información de cada defecto en el log de Registro de Defectos son precisos y completos – Verificar que los números de defectos introducidos y removidos por fase son razonables y correctos. – Corregir cualquier información de defecto omitido o incorrecto.
3	Tamaño	<ul style="list-style-type: none"> – Contabilizar el tamaño del programa completado – Determinar el tamaño de la código base, eliminado, modificado, reusado, total, y nuevo reusable – Ingresar esta información en el formulario Resumen del Plan del Proyecto .
4	Tiempo	<ul style="list-style-type: none"> – Revisar el log de Registro de Tiempos completado para ver si existen errores u omisiones – Corregir cualquier información de tiempo omitido o incompleto.

Criterios de Salida	<ul style="list-style-type: none"> – Un programa probado completamente de acuerdo a los Estándares de Codificación y conteo del tamaño – Formulario Resumen del Plan del Proyecto – Formulario PIP completado describiendo problemas del proceso, sugerencias de mejora y lecciones aprendidas – Log de Registros de Tiempos y Defectos completados
----------------------------	---

** Verifique que ha cumplido con las condiciones de salida de la fase de planificación. Los criterios de entrada ya se habían definido anteriormente en la sección de Planeación.

1. Registro de defectos

En total se encontraron 204 defectos, los cuales para corregir tomaron un tiempo total de 15 horas y 54 minutos

Defectos - Fase de Diseño							
Nro	Descripción	Tipo	Fase inyección	Modulo Inyeccion	Fase remoción	Módulo Remoción	Tiempo en solucionarlo
2	diagramación casos de uso (mensajes)		Diseño		Revisión diseño		0:32:00
1	tipos de flechas casos de uso		Diseño		Revisión diseño		0:12:00
1	copys registro o ingreso caso de uso		Diseño		Revisión diseño		0:01:00
1	soluciona transferencia horas extiende de ver ofertas		Diseño		Revisión diseño		0:02:00
1	Caso de uso que extiende de eliminar mensaje		Diseño		Revisión diseño		0:01:00
1	Cambio copys precondition casos de uso		Diseño		Revisión diseño		0:01:00
1	Elimina curso alternativo que no aplica en caso de uso 27		Diseño		Revisión diseño		0:01:00
8							0:50:00

Tabla 7. Registro de defectos - Diseño

Defectos - Fase de Desarrollo							
Nro	Descripción	Tipo	Fase inyección	Modulo Inyeccion	Fase remoción	Módulo Remoción	Tiempo en solucionar
1	Configuración de idiomas	No aplica	Planeación		Desarrollo	Instalación dependencias del proyecto (Django, postgresql)	0:12:00
1	Fixture para crear superusuario	No aplica	Planeación		Desarrollo	Configuración inicial del proyecto	0:15:00
1	Implementar listado de propuestas propias del usuario	No aplica	Planeación		Desarrollo	Implementación listado de propuestas de usuario	0:01:00

1	Eliminación de propuestas y ofertas	No aplica	Planeación		Desarrollo	Implementación opción de hacer y editar ofertas a las propuestas	0:02:00
2	Corrección importación de dependencias	Asignación	Desarrollo	Implementación registro de usuarios y entidades	Revisión desarrollo	Implementación registro de usuarios y entidades	0:01:00
3	Corrección nombres de variables enviadas en registro	Asignación	Desarrollo	Implementación registro de usuarios y entidades	Pruebas	Implementación registro de usuarios y entidades	0:03:00
3	Error en guardado de datos en modelo	Función	Desarrollo	Implementación registro de usuarios y entidades	Pruebas	Implementación registro de usuarios y entidades	0:09:00
1	Corrección importación de dependencias	Asignación	Desarrollo	Implementación login de usuarios	Revisión desarrollo	Implementación login de usuarios	0:01:00
2	Error en validación de usuario logueado	Función	Desarrollo	Implementación login de usuarios	Pruebas	Implementación login de usuarios	0:06:00
2	Organización en error de formulario	Función	Desarrollo	Implementación opción de cambiar contraseña	Pruebas	Implementación opción de cambiar contraseña	0:09:00
1	Imagen en header si no hay ninguna y url en perfil usuario	Entorno	Desarrollo	Implementación menu, footer generales del usuario	Desarrollo	Implementación opción de actualizar usuario	0:20:00
1	Campo definido en el serializador que no era (get_initial_skills)	Asignación	Desarrollo	Implementación opción de actualizar usuario	Revisión desarrollo	Implementación opción de actualizar usuario	0:01:00
1	Variable sin s al final usada en la plantilla	Asignación	Desarrollo	Implementación opción de actualizar usuario	Revisión desarrollo	Implementación opción de actualizar usuario	0:01:00
1	Validación máximo de caracteres descripción del usuario	Chequeo	Desarrollo	Implementación opción de actualizar usuario	Revisión desarrollo	Implementación opción de actualizar usuario	0:01:00
1	Nombre llave foránea de perfil de usuario mal escrita	Asignación	Desarrollo	Implementación opción de actualizar usuario	Revisión desarrollo	Implementación opción de actualizar usuario	0:01:00
1	Variable mal nombrada en template para perfil de usuario	Asignación	Desarrollo	Implementación opción de actualizar usuario	Revisión desarrollo	Implementación opción de actualizar usuario	0:02:00
1	Definición serializador como ListField (estaba DitchField)	Asignación	Desarrollo	Implementación opción de actualizar usuario	Compilación	Implementación opción de actualizar usuario	0:05:00
1	Definición errada de json en habilidades	Función	Desarrollo	Implementación opción de actualizar usuario	Compilación	Implementación opción de actualizar usuario	0:05:00
1	Importación template tag para carga	Asignación	Desarrollo	Implementación opción de	Compilación	Implementación opción de	0:01:00

	de archivos estáticos			actualizar usuario		actualizar usuario	
1	Estilos en botones de actualizar usuario (mal ubicados)	Entorno	Desarrollo	Implementación opción de actualizar usuario	Pruebas	Implementación opción de actualizar usuario	0:01:00
1	Clase mal nombrada en campo textarea	Asignación	Desarrollo	Implementación opción de actualizar usuario	Pruebas	Implementación opción de actualizar usuario	0:01:00
2	Error en conversión a json javascript desde python	Función	Desarrollo	Implementación opción de actualizar usuario	Pruebas	Implementación opción de actualizar usuario	0:20:00
1	Error en clase faltante en dirección	Entorno	Desarrollo	Implementación opción de actualizar usuario	Pruebas	Implementación opción de actualizar usuario	0:01:00
2	Error parseando json al guardar en python	Función	Desarrollo	Implementación opción de actualizar usuario	Pruebas	Implementación opción de actualizar usuario	1:10:00
1	Value en textarea (no usa atributo value)	Asignación	Desarrollo	Implementación opción de actualizar usuario	Pruebas	Implementación opción de actualizar usuario	0:01:00
1	error en contenedor de mensajes, debe instanciarse 1 vez	Asignación	Desarrollo	Implementación mensajes de alerta	Pruebas	Implementación mensajes de alerta	0:06:00
1	Organización visual de cards en el perfil de usuario	Entorno	Desarrollo	Implementación vista de perfil de un usuario	Pruebas	Implementación vista de perfil de un usuario	0:03:00
1	Error en dependencia de Postgres (se usa una versión anterior)	Construir, paquetes	Desarrollo	Implementación listado de propuestas de usuario	Revisión desarrollo	Implementación listado de propuestas de usuario	0:11:00
3	Sobreescribe estilos de dos librerías css	Entorno	Desarrollo	Implementación listado de propuestas de usuario	Revisión desarrollo	Implementación listado de propuestas de usuario	0:24:00
7	No se importaron dependencias javascript para ser usadas	Asignación	Desarrollo	Implementación listado de propuestas de usuario	Pruebas	Implementación listado de propuestas de usuario	0:14:00
2	Nombre de variable errada (state en vez de status en propuesta)	Asignación	Desarrollo	Implementación listado de propuestas de usuario	Pruebas	Implementación listado de propuestas de usuario	0:02:00
6	No se importaron dependencias javascript para ser usadas	Asignación	Desarrollo	Implementación creación de propuestas	Revisión desarrollo	Implementación creación de propuestas	0:07:00
1	Error en función que muestra mensajes de error de acuerdo al nombre del campo	Función	Desarrollo	Implementación creación de propuestas	Revisión desarrollo	Implementación creación de propuestas	0:24:00
1	Llamado de función equivocada (ciclo infinito)	Función	Desarrollo	Implementación listado de propuestas de usuario	Pruebas	Implementación creación de propuestas	0:24:00

9	No se importaron dependencias javascript para ser usadas	Asignación	Desarrollo	Implementación creación de propuestas	Pruebas	Implementación creación de propuestas	0:17:00
1	Error al utilizar componente, se hizo llamado como la version desactualizada y no como la ultima instalada en el proyecto (React Router)	Construir, paquetes	Desarrollo	Implementación creación de propuestas	Pruebas	Implementación creación de propuestas	0:13:00
1	Valida nombre de habilidad para ser creada como nueva, (se tomaba int como string)	Función	Desarrollo	Implementación creación de propuestas	Pruebas	Implementación creación de propuestas	0:17:00
1	Error en concatenar cadenas	Función	Desarrollo	Implementación creación de propuestas	Pruebas	Implementación creación de propuestas	0:02:00
3	Listado de campos sin coma (JSON)	Sintaxis	Desarrollo	Implementación creación de propuestas	Pruebas	Implementación creación de propuestas	0:03:00
1	No se realizó metodo get en vista de propuestas	Asignación	Desarrollo	Implementación vista de propuesta como página	Revisión desarrollo	Implementación vista de propuesta como página	0:03:00
1	Definición de objeto como lista	Función	Desarrollo	Implementación vista de propuesta como página	Revisión desarrollo	Implementación vista de propuesta como página	0:01:00
2	No se importó dependencia en react	Asignación	Desarrollo	Implementación vista de propuesta como página	Revisión desarrollo	Implementación vista de propuesta como página	0:01:00
1	Nombre de variable errada (proposals en vez de proposal)	Asignación	Desarrollo	Implementación vista de propuesta como página	Revisión desarrollo	Implementación vista de propuesta como página	0:01:00
1	Encerrar componente en un solo tag html	Función	Desarrollo	Implementación vista de propuesta como página	Pruebas	Implementación vista de propuesta como página	0:02:00
1	Definición de variable vacía errada ({} en vez de null)	Función	Desarrollo	Implementación vista de propuesta como página	Pruebas	Implementación vista de propuesta como página	0:09:00
1	Propiedad usada pero inexistente en componente de react (spacing en Grid)	Construir, paquetes	Desarrollo	Implementación vista de propuesta como página	Pruebas	Implementación vista de propuesta como página	0:01:00
1	Error en importación de componente de usuario (se importaba como si	Asignación	Desarrollo	Implementación vista de propuesta como página	Pruebas	Implementación vista de propuesta como página	0:51:00

	fuera de propuesta)						
1	Adiciona llave (key) en iteración de habilidades	Función	Desarrollo	Implementación vista de propuesta como página	Pruebas	Implementación vista de propuesta como página	0:01:00
1	Uso de clases de manera equivocada	Función	Desarrollo	Implementación vista de propuesta como página	Pruebas	Implementación vista de propuesta como página	0:06:00
1	Retorna Nulo en vez de vacío ({}), no se validaba correctamente	Sintaxis	Desarrollo	Implementación vista de propuesta como página	Pruebas	Implementación vista de propuesta como página	0:07:00
1	Url llamada erróneamente, hacia match con una url definida anteriormente	Asignación	Desarrollo	Implementación vista de propuesta como página	Pruebas	Implementación vista de propuesta como página	0:02:00
1	No se colocaba en vacío los datos del form al intentar crear una propuesta después de seleccionar actualizar propuesta	Función	Desarrollo	Implementación actualización de propuestas	Revisión desarrollo	Implementación actualización de propuestas	0:04:00
1	Encierra variable en llaves ({} reactjs)	Sintaxis	Desarrollo	Implementación actualización de propuestas	Pruebas	Implementación actualización de propuestas	0:01:00
1	Quita llaves en variable ({} reactjs)	Sintaxis	Desarrollo	Implementación actualización de propuestas	Pruebas	Implementación actualización de propuestas	0:01:00
1	Nombre errado en llamado de función	Asignación	Desarrollo	Implementación actualización de propuestas	Pruebas	Implementación actualización de propuestas	0:09:00
1	Parsea listado habilidades de acuerdo al formato del componente (value, label)	Función	Desarrollo	Implementación actualización de propuestas	Pruebas	Implementación actualización de propuestas	0:10:00
1	No se importó método (putData)	Asignación	Desarrollo	Implementación actualización de propuestas	Pruebas	Implementación actualización de propuestas	0:01:00
1	Adiciona slash al final de la url en actualizar	Sintaxis	Desarrollo	Implementación actualización de propuestas	Pruebas	Implementación actualización de propuestas	0:02:00
1	Validación de propuesta para crear o editar equivocada	Chequeo	Desarrollo	Implementación actualización de propuestas	Pruebas	Implementación actualización de propuestas	0:08:00
1	Nuevos skills no son guardados en el sistema al actualizar propuesta	Función	Desarrollo	Implementación actualización de propuestas	Pruebas	Implementación actualización de propuestas	0:05:00
2	Soluciona loop infinito en función de	Función	Desarrollo	Implementación actualización	Pruebas	Implementación actualización	0:25:00

	convertir tiempo a horas			de propuestas		de propuestas	
1	Error en validación de habilidades nuevas (se recibían como string, cuando debía ser diccionario)	Función	Desarrollo	Implementación actualización de propuestas	Pruebas	Implementación actualización de propuestas	0:38:00
1	Error en validación del campo habilidades (si se enviaba vacío no se actualizaba en base de datos)	Función	Desarrollo	Implementación actualización de propuestas	Pruebas	Implementación actualización de propuestas	0:12:00
1	Soluciona validación de horas restantes disponibles después de crear o actualizar propuestas	Función	Desarrollo	Implementación actualización de propuestas	Pruebas	Implementación actualización de propuestas	0:16:00
1	Elimina importación de método inexistente	Asignación	Desarrollo	Implementación buscador y filtro (lugar, habilidades, horas) de propuestas	Revisión desarrollo	Implementación buscador y filtro (lugar, habilidades, horas) de propuestas	0:01:00
1	Cierra paréntesis en función	Sintaxis	Desarrollo	Implementación buscador y filtro (lugar, habilidades, horas) de propuestas	Revisión desarrollo	Implementación buscador y filtro (lugar, habilidades, horas) de propuestas	0:01:00
2	Soluciona comas faltantes en json definido para campos del formulario	Sintaxis	Desarrollo	Implementación buscador y filtro (lugar, habilidades, horas) de propuestas	Revisión desarrollo	Implementación buscador y filtro (lugar, habilidades, horas) de propuestas	0:01:00
1	Soluciona importación de la librería re (usada para expresiones regulares)	Asignación	Desarrollo	Implementación buscador y filtro (lugar, habilidades, horas) de propuestas	Pruebas	Implementación buscador y filtro (lugar, habilidades, horas) de propuestas	0:01:00
1	Añade función para actualización de campos al editar (setState)	Función	Desarrollo	Implementación buscador y filtro (lugar, habilidades, horas) de propuestas	Pruebas	Implementación buscador y filtro (lugar, habilidades, horas) de propuestas	0:01:00
1	Soluciona definición de variable (estaba como constante y era variable)	Asignación	Desarrollo	Implementación buscador y filtro (lugar, habilidades, horas) de propuestas	Pruebas	Implementación buscador y filtro (lugar, habilidades, horas) de propuestas	0:01:00
1	Reorganiza definición de variable (se definía más abajo del lugar donde se llamaba)	Asignación	Desarrollo	Implementación buscador y filtro (lugar, habilidades, horas) de propuestas	Pruebas	Implementación buscador y filtro (lugar, habilidades, horas) de propuestas	0:01:00
1	Redefine formulario de búsqueda como componente y no como función	Función	Desarrollo	Implementación buscador y filtro (lugar, habilidades,	Pruebas	Implementación buscador y filtro (lugar, habilidades,	0:07:00

				horas) de propuestas		horas) de propuestas	
1	Soluciona llamado de variable (faltaba this.props)	Sintaxis	Desarrollo	Implementación buscador y filtro (lugar, habilidades, horas) de propuestas	Pruebas	Implementación buscador y filtro (lugar, habilidades, horas) de propuestas	0:01:00
3	Convierte entero en string para concatenar	Función	Desarrollo	Implementación buscador y filtro (lugar, habilidades, horas) de propuestas	Pruebas	Implementación buscador y filtro (lugar, habilidades, horas) de propuestas	0:01:00
1	Elimina uso de range (python) en paginación (no es permitido por la API)	Función	Desarrollo	Implementación buscador y filtro (lugar, habilidades, horas) de propuestas	Pruebas	Implementación buscador y filtro (lugar, habilidades, horas) de propuestas	0:01:00
1	Soluciona envío de datos en submit del formulario de búsqueda	Función	Desarrollo	Implementación buscador y filtro (lugar, habilidades, horas) de propuestas	Pruebas	Implementación buscador y filtro (lugar, habilidades, horas) de propuestas	0:01:00
1	Soluciona importación de componente	Asignación	Desarrollo	Implementación buscador y filtro (lugar, habilidades, horas) de propuestas	Pruebas	Implementación buscador y filtro (lugar, habilidades, horas) de propuestas	0:01:00
1	Soluciona nombre equivocado usado en el campo de buscar (name)	Asignación	Desarrollo	Implementación buscador y filtro (lugar, habilidades, horas) de propuestas	Pruebas	Implementación buscador y filtro (lugar, habilidades, horas) de propuestas	0:01:00
1	Soluciona error en datos por defecto vacíos en componentes (Default props)	Asignación	Desarrollo	Implementación buscador y filtro (lugar, habilidades, horas) de propuestas	Pruebas	Implementación buscador y filtro (lugar, habilidades, horas) de propuestas	0:01:00
1	Nombre de clase mal escrito	Asignación	Desarrollo	Implementación opción de hacer y editar ofertas a las propuestas	Revisión desarrollo	Implementación opción de hacer y editar ofertas a las propuestas	0:01:00
1	Serializador importado con nombre mal escrito	Asignación	Desarrollo	Implementación opción de hacer y editar ofertas a las propuestas	Revisión desarrollo	Implementación opción de hacer y editar ofertas a las propuestas	0:01:00
1	Filtro de ofertas retornaba una lista, cuando debía ser un objeto	Función	Desarrollo	Implementación opción de hacer y editar ofertas a las propuestas	Revisión desarrollo	Implementación opción de hacer y editar ofertas a las propuestas	0:01:00
1	Importación circular de modelos de ofertas y propuestas	Asignación	Desarrollo	Implementación opción de hacer y editar ofertas a las	Pruebas	Implementación opción de hacer y editar ofertas a las	0:01:00

				propuestas		propuestas	
2	Importación circular de serializadores de ofertas y propuestas	Asignación	Desarrollo	Implementación opción de hacer y editar ofertas a las propuestas	Pruebas	Implementación opción de hacer y editar ofertas a las propuestas	0:01:00
1	Soluciona importar componente	Función	Desarrollo	Implementación opción de hacer y editar ofertas a las propuestas	Pruebas	Implementación opción de hacer y editar ofertas a las propuestas	0:01:00
1	Soluciona formato equivocado de estilos	Entorno	Desarrollo	Implementación opción de hacer y editar ofertas a las propuestas	Pruebas	Implementación opción de hacer y editar ofertas a las propuestas	0:01:00
1	Soluciona formato de renderizado de react (se usó paréntesis en vez de llaves)	Sintaxis	Desarrollo	Implementación opción de hacer y editar ofertas a las propuestas	Pruebas	Implementación opción de hacer y editar ofertas a las propuestas	0:50:00
1	Se Corrige nombre de función	Asignación	Desarrollo	Implementación opción de hacer y editar ofertas a las propuestas	Pruebas	Implementación opción de hacer y editar ofertas a las propuestas	0:01:00
1	Organiza función como exportable en javascript (export)	Sintaxis	Desarrollo	Implementación opción de hacer y editar ofertas a las propuestas	Pruebas	Implementación opción de hacer y editar ofertas a las propuestas	0:01:00
1	Soluciona importación de componente faltante	Asignación	Desarrollo	Implementación opción de hacer y editar ofertas a las propuestas	Pruebas	Implementación opción de hacer y editar ofertas a las propuestas	0:01:00
1	Importa componente correctamente (se usó llaves, no eran necesarias)	Sintaxis	Desarrollo	Implementación opción de hacer y editar ofertas a las propuestas	Pruebas	Implementación opción de hacer y editar ofertas a las propuestas	0:45:00
1	Soluciona importación de función	Asignación	Desarrollo	Implementación opción de hacer y editar ofertas a las propuestas	Pruebas	Implementación opción de hacer y editar ofertas a las propuestas	0:01:00
1	Soluciona serializador usado de un modelo que no era (user en vez de userprofile)	Asignación	Desarrollo	Implementación opción de hacer y editar ofertas a las propuestas	Pruebas	Implementación opción de hacer y editar ofertas a las propuestas	0:01:00
1	Adiciona id en serializador de ofertas	Asignación	Desarrollo	Implementación opción de hacer y editar ofertas a las	Pruebas	Implementación opción de hacer y editar ofertas a las	0:01:00

				propuestas		propuestas	
1	Soluciona nombre de variable, se quita s al final	Asignación	Desarrollo	Implementación opción de hacer y editar ofertas a las propuestas	Pruebas	Implementación opción de hacer y editar ofertas a las propuestas	0:01:00
1	Soluciona definición de variable y se pasa a una función	Función	Desarrollo	Implementación opción de eliminar propuestas y ofertas	Revisión desarrollo	Implementación opción de eliminar propuestas y ofertas	0:01:00
1	Adiciona llamado de función para eliminar en el botón de eliminar	Función	Desarrollo	Implementación opción de eliminar propuestas y ofertas	Revisión desarrollo	Implementación opción de eliminar propuestas y ofertas	0:02:00
1	Importa componente de alerta antes de eliminar	Función	Desarrollo	Implementación opción de eliminar propuestas y ofertas	Revisión desarrollo	Implementación opción de eliminar propuestas y ofertas	0:01:00
1	Adiciona coma faltante en json	Sintaxis	Desarrollo	Implementación opción de eliminar propuestas y ofertas	Revisión desarrollo	Implementación opción de eliminar propuestas y ofertas	0:01:00
1	Corrige nombre de función (mal llamada)	Asignación	Desarrollo	Implementación opción de eliminar propuestas y ofertas	Pruebas	Implementación opción de eliminar propuestas y ofertas	0:02:00
2	Elimina variable sin usar en función (ev)	Asignación	Desarrollo	Implementación opción de eliminar propuestas y ofertas	Pruebas	Implementación opción de eliminar propuestas y ofertas	0:01:00
2	Organiza parámetro faltante en llamado de api para eliminar	Sintaxis	Desarrollo	Implementación opción de eliminar propuestas y ofertas	Pruebas	Implementación opción de eliminar propuestas y ofertas	0:12:00
2	Cambia estatus de respuesta al eliminar propuesta u oferta desde el backend (el status que estaba enviaba null)	Función	Desarrollo	Implementación opción de eliminar propuestas y ofertas	Pruebas	Implementación opción de eliminar propuestas y ofertas	0:01:00
2	Adiciona código para cerrar mensajes de alerta (Alerts) después de dar click en aceptar o cancelar una oferta	Función	Desarrollo	Implementación opción de aceptar y rechazar ofertas a propuestas	Revisión desarrollo	Implementación opción de aceptar y rechazar ofertas a propuestas	0:01:00
1	Valida estado de una oferta (para solo permitir que se pueda rechazar una oferta una sola vez)	Función	Desarrollo	Implementación opción de aceptar y rechazar ofertas a propuestas	Revisión desarrollo	Implementación opción de aceptar y rechazar ofertas a propuestas	0:02:00
1	Ingresa componente de mensajes de alerta (Alerts) dentro de div al hacer render	Sintaxis	Desarrollo	Implementación opción de aceptar y rechazar ofertas a propuestas	Pruebas	Implementación opción de aceptar y rechazar ofertas a propuestas	0:01:00
1	Soluciona error al concatenar (no se	Sintaxis	Desarrollo	Implementación opción de	Pruebas	Implementación opción de	0:01:00

	encerró el string con la variable en {})			aceptar y rechazar ofertas a propuestas		aceptar y rechazar ofertas a propuestas	
4	Importa componentes de Alert, putData, Notification y Notification manager en componente de ofertas	Asignación	Desarrollo	Implementación opción de aceptar y rechazar ofertas a propuestas	Pruebas	Implementación opción de aceptar y rechazar ofertas a propuestas	0:01:00
1	Corrige nombre de variable (era otro nombre)	Asignación	Desarrollo	Implementación opción de aceptar y rechazar ofertas a propuestas	Pruebas	Implementación opción de aceptar y rechazar ofertas a propuestas	0:03:00
1	Corrige estado de propuesta, no se actualizaba al ser aceptada una oferta	Función	Desarrollo	Implementación opción de aceptar y rechazar ofertas a propuestas	Pruebas	Implementación opción de aceptar y rechazar ofertas a propuestas	0:08:00
1	Soluciona error en validacion del serializador de ofertas, se valida que solo sea al actualizar (estaba al crear también)	Función	Desarrollo	Implementación opción de aceptar y rechazar ofertas a propuestas	Pruebas	Implementación opción de aceptar y rechazar ofertas a propuestas	0:06:00
1	Soluciona error en validación de sentencia (if else)	Sintaxis	Desarrollo	Implementación finalización propuestas y transferencias de horas	Revisión desarrollo	Implementación finalización propuestas y transferencias de horas	0:01:00
1	Elimina espacio de más al hacer llamado de un array	Sintaxis	Desarrollo	Implementación finalización propuestas y transferencias de horas	Revisión desarrollo	Implementación finalización propuestas y transferencias de horas	0:01:00
1	Se corrige variable que no estaba definida	Asignación	Desarrollo	Implementación finalización propuestas y transferencias de horas	Pruebas	Implementación finalización propuestas y transferencias de horas	0:11:00
1	Se corrige dependencia no importada	Asignación	Desarrollo	Implementación finalización propuestas y transferencias de horas	Pruebas	Implementación finalización propuestas y transferencias de horas	0:01:00
1	Variable usada no importada en función	Asignación	Desarrollo	Implementación finalización propuestas y transferencias de horas	Pruebas	Implementación finalización propuestas y transferencias de horas	0:01:00
1	Variable no estaba definida (se definió usando let)	Sintaxis	Desarrollo	Implementación finalización propuestas y transferencias de horas	Pruebas	Implementación finalización propuestas y transferencias de horas	0:01:00

1	Corrige texto en componente de diálogo (error tag p dentro de tag p)	Sintaxis	Desarrollo	Implementación finalización propuestas y transferencias de horas	Pruebas	Implementación finalización propuestas y transferencias de horas	0:07:00
1	Corrige validación en serializadores (validaba al crear objetos solamente y no al editar)	Función	Desarrollo	Implementación finalización propuestas y transferencias de horas	Pruebas	Implementación finalización propuestas y transferencias de horas	0:03:00
1	Corrige opciones en selector en horas adicionales de propuestas (se enviaba vacío)	Función	Desarrollo	Implementación finalización propuestas y transferencias de horas	Pruebas	Implementación finalización propuestas y transferencias de horas	0:06:00
1	Valida que al aceptar una oferta a una propuesta se oculten los botones de poder aceptar a las demás	Función	Desarrollo	Implementación finalización propuestas y transferencias de horas	Pruebas	Implementación finalización propuestas y transferencias de horas	0:06:00
1	Reduce horas disponibles de un usuario al aceptar una oferta	Función	Desarrollo	Implementación finalización propuestas y transferencias de horas	Pruebas	Implementación finalización propuestas y transferencias de horas	0:21:00
1	Se corrige número entero enviado como string en componente	Sintaxis	Desarrollo	Implementación finalización propuestas y transferencias de horas	Pruebas	Implementación finalización propuestas y transferencias de horas	0:01:00
2	Se corrige error en nombres de los campos del formulario	Función	Desarrollo	Implementación envío de mensajes entre usuarios	Pruebas	Implementación envío de mensajes entre usuarios	0:03:00
1	Se corrige nombre de variable en lista de mensajes	Asignación	Desarrollo	Implementación envío de mensajes entre usuarios	Pruebas	Implementación envío de mensajes entre usuarios	0:01:00
1	Corrige nombre de variable (era otro nombre)	Asignación	Desarrollo	Implementación dashboard de notificaciones de mensajes	Pruebas	Implementación dashboard de notificaciones de mensajes	0:01:00
2	Variable de función llamada equivocadamente	Función	Desarrollo	Implementación calificaciones entre usuarios	Revisión desarrollo	Implementación calificaciones entre usuarios	0:01:00
1	Llama función en componente que no existía y no era necesario	Asignación	Desarrollo	Implementación calificaciones entre usuarios	Revisión desarrollo	Implementación calificaciones entre usuarios	0:01:00
1	Importación de componente con nombre equivocado	Asignación	Desarrollo	Implementación calificaciones entre usuarios	Pruebas	Implementación calificaciones entre usuarios	0:01:00
2	Concatenación de variables mal nombrada	Sintaxis	Desarrollo	Implementación calificaciones entre usuarios	Pruebas	Implementación calificaciones entre usuarios	0:04:00

1	Elementos adyacentes no estaban encerrados en un tag (reactjs)	Sintaxis	Desarrollo	Implementación calificaciones entre usuarios	Pruebas	Implementación calificaciones entre usuarios	0:03:00
1	Librería importada equivocadamente	Sintaxis	Desarrollo	Implementación calificaciones entre usuarios	Pruebas	Implementación calificaciones entre usuarios	0:01:00
1	Elementos adyacentes no estaban encerrados en un tag (reactjs)	Sintaxis	Desarrollo	Implementación calificaciones entre usuarios	Pruebas	Implementación calificaciones entre usuarios	0:12:00
1	Paréntesis en vez de {} en ciclo for	Sintaxis	Desarrollo	Implementación calificaciones entre usuarios	Pruebas	Implementación calificaciones entre usuarios	0:09:00
1	Corrección de serialización (json) de método get_queryset	Asignación	Desarrollo	Implementación calificaciones entre usuarios	Pruebas	Implementación calificaciones entre usuarios	0:07:00
1	Soluciona return en uso de método .map (javascript)	Sintaxis	Desarrollo	Implementación calificaciones entre usuarios	Pruebas	Implementación calificaciones entre usuarios	0:03:00
2	Soluciona importación de dependencia	Asignación	Desarrollo	Implementación calificaciones entre usuarios	Pruebas	Implementación calificaciones entre usuarios	0:01:00
1	Error en variable no declarada en método render (reacts)	Asignación	Desarrollo	Implementación calificaciones entre usuarios	Pruebas	Implementación calificaciones entre usuarios	0:01:00
1	Soluciona función llamada equivocadamente	Función	Desarrollo	Implementación calificaciones entre usuarios	Pruebas	Implementación calificaciones entre usuarios	0:02:00
1	Elimina variable innecesaria en función	Asignación	Desarrollo	Implementación calificaciones entre usuarios	Pruebas	Implementación calificaciones entre usuarios	0:01:00
1	Corrige envío de variable mal enviada en serializador	Asignación	Desarrollo	Implementación calificaciones entre usuarios	Pruebas	Implementación calificaciones entre usuarios	0:01:00
2	Corrige nombre de variable, primera letra en minúscula	Asignación	Desarrollo	Implementación calificaciones entre usuarios	Pruebas	Implementación calificaciones entre usuarios	0:05:00
2	Corrige error en paginador	Función	Desarrollo	Implementación calificaciones entre usuarios	Pruebas	Implementación calificaciones entre usuarios	0:50:00
1	Corrige validación al editar propuestas	Chequeo	Desarrollo	Implementación creación de propuestas	Pruebas	Implementación calificaciones entre usuarios	0:04:00
1	Corrige importación de dependencia	Asignación	Desarrollo	Integración calificaciones al perfil de usuario en propuesta y ofertas	Revisión desarrollo	Integración calificaciones al perfil de usuario en propuesta y ofertas	0:01:00

2	Corrige forma de importar variable (labels)	Sintaxis	Desarrollo	Integración calificaciones al perfil de usuario en propuesta y ofertas	Pruebas	Integración calificaciones al perfil de usuario en propuesta y ofertas	0:16:00
1	Error en retorno de errores al actualizar usuario, se accedía a variable inexistente	Función	Desarrollo	Implementación opción de actualizar usuario	Pruebas	Integración calificaciones al perfil de usuario en propuesta y ofertas	0:04:00
3	Corrige uso de variable namespace en urls (no era necesario)	Sintaxis	Desarrollo	Implementación términos y condiciones del proyecto y aceptación en el registro	Pruebas	Implementación términos y condiciones del proyecto y aceptación en el registro	0:02:00
1	Se corrige validación en correcta al hacer submit del formulario	Función	Desarrollo	Implementación términos y condiciones del proyecto y aceptación en el registro	Pruebas	Implementación términos y condiciones del proyecto y aceptación en el registro	0:11:00
204							15:54:00

Tabla 8. Registro de defectos- Desarrollo

2. Tipos de defectos

Tipos de defectos			
Tipo	Nombre	Cantidad	Tiempo en solucionar (horas)
10	Documentación	0	0:00:00
20	Sintaxis	36	3:09:00
30	Construir, paquetes	3	0:25:00
40	Asignación	89	3:10:00
50	Interfaz	0	0:00:00
60	Chequeo	3	0:13:00
70	Datos	0	0:00:00
80	Función	61	7:37:00
90	Sistema	0	0:00:00
100	Entorno	8	0:50:00
Totales		200	15:24:00

Tabla 9. Tipos de defectos

En la fase de planeación se insertaron 4 defectos que es difícil encajar en alguno de estos tipos, estos defectos suman 30 minutos.

Los tipos de defectos de Asignación y Sintaxis deberían haberse detectado en su mayoría desde las revisiones de código, casi todos los defectos de asignación fueron por no importar las dependencias requeridas o por estar mal nombradas, defectos que se debieron hallar desde la misma codificación o desde una revisión de código más minuciosa, estos representan el casi el 62% de todos las fallas encontradas.

3. Resumen del Plan del proyecto y de defectos

Resumen del Plan del proyecto		
Tiempos por fase		
Fase	Tiempo planeado	Tiempo actual
Planeación	22:30:00	22:03:34
Diseño	17:00:00	18:50:00
Revisión Diseño	6:30:00	2:11:00
Codificación	67:30:00	92:27:00
Revisión Codificación	22:00:00	5:52:00
Compilación	6:00:00	6:23:00
Pruebas	19:00:00	14:04:00
Postmortem	10:00:00	13:20:00
Total	168 horas	165 horas, 40 minutos, 34 segundos

Tabla 10. Resumen del Plan del proyecto

Defectos inyectados		
Fase	Planeados	Actuales
Planeación	2	4
Diseño	2	8
Revisión Diseño	0	0
Codificación	20	200
Revisión Codificación	0	0
Compilación	10	0
Pruebas	0	0
Postmortem	0	0
Total	34	212

Tabla 11. Resumen de defectos inyectados

Defectos removidos		
Fase	Planeados	Actuales
Planeación	0	0
Diseño	0	0
Revisión Diseño	0	8
Codificación	4	4
Revisión Codificación	0	45
Compilación	20	125
Pruebas	10	30
Postmortem	0	0
Total	34	212

Tabla 12. Resumen de defectos removidos

4. Productividad de la aplicación

La productividad se calcula con la siguiente fórmula:

$$Productividad = \frac{Total\ de\ Líneas\ de\ Código}{Tiempo\ total}$$

$$Productividad = \frac{5.667}{165.43}$$

La productividad tomando solo los lenguajes de programación fue de: **34 LOC/Hr** aproximadamente.

La productividad sumando la estructura en HTML y estilos (CSS) fue de: **49 LOC/Hr** Aproximadamente.

5. Calidad de la aplicación

La calidad de la aplicación debe partir de una responsabilidad personal, de mantener una disciplina en el desarrollo y en el seguimiento de los planes realizados. La calidad es una prioridad máxima.

5.1. Principios de calidad

PSP ha definido algunos principios de calidad del software, como los siguientes:

- Es menos costoso corregir defectos antes en un proceso, que después.
- Cuanto más tiempo permanece un defecto en un producto, mayor será el costo para extraerlo
- Las pruebas son una manera ineficiente e ineficaz para eliminar defectos.
- Las revisiones son más eficientes que las pruebas para corregir defectos.
- Es más eficiente prevenir defectos que corregirlos.

PSP hace mucho énfasis en las revisiones de código, no solo en hacer las revisiones, sino en hacerlas bien, en destinar un porcentaje de tiempo del desarrollo para hacerlas y para detectar defectos antes de pasar a compilación y pruebas.

5.2. Métricas de calidad

Las métricas de calidad son útiles para comprender y refinar el proceso personal.

5.2.1. Yield de fase

$$\text{yield de fase} = \frac{\text{Número de defectos removidos en la fase} \times 100}{\text{Número de defectos inyectados en la fase} + \text{Número de defectos inyectados remanentes}}$$

Revisión de diseño:

$$yield = \frac{8 \times 100}{0 + 12}$$

El porcentaje de defectos removidos durante la revisión de diseño fue de: **66.67 %**

Codificación:

$$yield = \frac{4 \times 100}{0 + 4}$$

El porcentaje de defectos removidos durante la fase de codificación fue de: **100 %**

Revisión de codificación:

$$yield = \frac{45 \times 100}{200 + 0}$$

El porcentaje de defectos removidos durante la fase de revisión de codificación fue de: **22.5 %**

Compilación:

$$yield = \frac{125 \times 100}{0 + 155}$$

El porcentaje de defectos removidos durante la fase de revisión de codificación fue de: **80.65 %**

5.2.2. Yield del proceso

$$yield \text{ del proceso} = \frac{\text{Número de defectos removidos antes de Compilación y Pruebas} \times 100}{\text{Número de defectos inyectados antes de Compilación y Pruebas}}$$

$$\text{yield del proceso} = \frac{57 \times 100}{212}$$

El porcentaje de defectos removidos antes de entrar en la fase de compilación o pruebas fue de: **26.89 %**

5.2.3. Yield de revisión

$$\text{yield del revisión} = \frac{\text{Número de defectos removidos en revisiones} \times 100}{\text{Número de defectos inyectados antes de Compilación y Pruebas}}$$

$$\text{yield del revisión} = \frac{53 \times 100}{212}$$

El porcentaje de defectos removidos durante las revisiones fue de: **25 %**

5.2.4. Porcentaje de costo de evaluación de la calidad COQ

$$\% \text{ Evaluación COQ} = \frac{\text{Tiempo de Revisión de Diseño y Código} \times 100}{\text{Tiempo de desarrollo total}}$$

$$\% \text{ Evaluación COQ} = \frac{(101 \text{ minutos} + 352 \text{ minutos}) \times 100}{9940,34 \text{ minutos}}$$

El porcentaje de tiempo de desarrollo empleado en revisión de diseño y codificación fue de: **4.55 %**

5.2.5. Porcentaje de fallas COQ

$$\% \text{ Fallas COQ} = \frac{\text{Tiempo de Compilación y Pruebas} \times 100}{\text{Tiempo de desarrollo total}}$$

$$\% \text{ Fallas COQ} = \frac{(383 \text{ minutos} + 844 \text{ minutos}) \times 100}{9940,34 \text{ minutos}}$$

El porcentaje de tiempo de desarrollo empleado en compilación y pruebas fue de:
12.34 %

5.2.6. Costo de la calidad COQ

$$\text{Costo de Calidad COQ} = \% \text{ Evaluación COQ} + \% \text{ Fallas COQ}$$

$$\text{Costo de Calidad COQ} = 4.55\% + 12.34\%$$

El porcentaje de tiempo dedicado a realizar tareas de evaluación y corrección fue de: **16.89 %**.

5.2.7. Relación de evaluación / fallas COQ

$$\text{COQ \% Evaluación} / \% \text{ Fallas} = \frac{\% \text{ Evaluación COQ}}{\% \text{ Fallas COQ}}$$

$$\text{COQ \% Evaluación} / \% \text{ Fallas} = \frac{4.55 \%}{12.34 \%}$$

La proporción de tiempo invertido en tareas de evaluación con respecto al tiempo invertido en tareas de corrección de fallas fue de: **0.37**

5.2.8. Densidad de defectos

Densidad de defectos totales:

$$\text{Densidad de defectos} = \frac{\text{Número de defectos encontrados} \times 1000}{\text{Unidad de tamaño}}$$

$$\text{Densidad de defectos} = \frac{212 \times 1000}{5.667 \text{ LOC}}$$

El número de defectos detectados por cada 1000 líneas de código fue de: **37.41**

Densidad de defectos en compilación:

$$\text{Densidad de defectos} = \frac{\text{Número de defectos encontrados en compilación} \times 1000}{\text{Unidad de tamaño}}$$

$$\text{Densidad de defectos} = \frac{125 \times 1000}{5.667 \text{ LOC}}$$

El número de defectos detectados por cada 1000 líneas de código fue de: **22.06**

Tomando en cuenta la estructura (HTML) y estilos (CSS3) la densidad de defectos en la fase de compilación fue de: **15.28**

Densidad de defectos en pruebas:

$$\text{Densidad de defectos} = \frac{\text{Número de defectos encontrados en pruebas} \times 1000}{\text{Unidad de tamaño}}$$

$$\text{Densidad de defectos} = \frac{30 \times 1000}{5.667 \text{ LOC}}$$

El número de defectos detectados por cada 1000 líneas de código fue de: **5.29**

Tomando en cuenta la estructura (HTML) y estilos (CSS3) la densidad de defectos en la fase de pruebas fue de: **3.67**

5.2.9. Índice de calidad del proceso (PQI)

Calidad del diseño:

$$\text{Calidad del diseño} = \frac{\text{Tiempo de diseño}}{\text{Tiempo de codificación}}$$

$$\text{Calidad del diseño} = \frac{925 \text{ minutos}}{5.547 \text{ minutos}}$$

La calidad del diseño fue de: **0.17**

Calidad de revisión del diseño:

$$\text{Calidad de revisión del diseño} = \frac{2 \times \text{Tiempo de revisión del diseño}}{\text{Tiempo del diseño}}$$

$$\text{Calidad de revisión del diseño} = \frac{2 \times 101 \text{ minutos}}{925 \text{ minutos}}$$

La Calidad de revisión del diseño fue de: **0.22**

Calidad de revisión del código:

$$\text{Calidad de revisión del código} = \frac{2 \times \text{Tiempo de revisión del código}}{\text{Tiempo del código}}$$

$$\text{Calidad de revisión del código} = \frac{2 \times 352 \text{ minutos}}{5.547 \text{ minutos}}$$

La Calidad de revisión del diseño fue de: **0.13**

Calidad de codificación:

$$\text{Calidad de codificación} = \frac{20}{10 + \text{densidad de defectos en compilación}}$$

$$\text{Calidad de codificación} = \frac{20}{10 + 15.28}$$

La Calidad de codificación fue de: **0.79**

Calidad del programa:

$$\text{Calidad de programa} = \frac{10}{5 + \text{densidad de defectos en pruebas}}$$

$$\text{Calidad de codificación} = \frac{10}{5 + 3.67}$$

La calidad de programa fue de: **1.15**

Calcular el PQI

Los 5 componentes de PQI se deben normalizar a [0, 1], "0" representa una mala práctica y "1" la práctica deseada.

$$PQI = 0.17 \times 0.22 \times 0.13 \times 0.79 \times 1$$

El PQI fue de: **0**

5.2.10. Tasa de eliminación de defectos de fase

$$\text{Tasa de eliminación de defectos de fase} = \frac{\text{Número de defectos removidos en la fase} \times 60}{\text{Tiempo de la fase (minutos)}}$$

Revisión de diseño:

$$\text{Tasa de eliminación de defectos de la revisión de diseño} = \frac{8 \times 60 \text{ min}}{101 \text{ min}}$$

El número de defectos encontrados por hora en la fase de revisión de diseño fue:
4.75

Revisión de codificación:

$$Tasa\ de\ eliminación\ de\ defectos\ de\ la\ revisión\ de\ codificación = \frac{45 \times 60\ min}{352\ min}$$

El número de defectos encontrados por hora en la fase de revisión de codificación fue: **7.67**

Compilación:

$$Tasa\ de\ eliminación\ de\ defectos\ de\ la\ compilación = \frac{125 \times 60\ min}{383\ min}$$

El número de defectos encontrados por hora en la fase de compilación fue: **18.58**

Pruebas:

$$Tasa\ de\ eliminación\ de\ defectos\ de\ las\ pruebas = \frac{30 \times 60\ min}{844\ min}$$

El número de defectos encontrados por hora en la fase de compilación fue: **2.13**

5.2.11. Tasa de revisión

$$Tasa\ de\ revisión = \frac{Tamaño\ del\ producto\ (LOC) \times 60}{Tiempo\ de\ la\ fase\ (minutos)}$$

Revisión de codificación:

$$Tasa\ de\ revisión\ de\ codificación = \frac{5.667\ LOC \times 60\ min}{383\ min}$$

El tamaño de producto revisado por hora es de: **887.78** Líneas de código.

5.2.12. Apalancamiento de eliminación de defectos (DLR)

$$DLR \text{ revisión de diseño} = \frac{\text{Tasa de eliminación de defectos de revisión de diseño}}{\text{Tasa de eliminación de defectos de pruebas}}$$

$$DLR \text{ revisión de diseño} = \frac{4.75}{2.13}$$

La medida de la eficacia relativa de eliminación de defectos en la revisión de diseño fue de: **2.23**

$$DLR \text{ revisión de codificación} = \frac{\text{Tasa de eliminación de defectos de revisión de codificación}}{\text{Tasa de eliminación de defectos de pruebas}}$$

$$DLR \text{ revisión de codificación} = \frac{7.67}{2.13}$$

La medida de la eficacia relativa de eliminación de defectos en la revisión de codificación fue de: **3.6**

6. Análisis de la Calidad

PSP hace hincapié en hacer revisiones de código que permitan encontrar defectos antes de las fases de compilación y/o pruebas, el yield de revisión fue del 25%, es decir que de todos los defectos que se corrigieron en el sistema solo se detectaron en las revisiones $\frac{1}{4}$ de ellos, un porcentaje que es bajo.

Con respecto al porcentaje de fallas (COQ), el resultado fue que más del 12% por ciento del tiempo fue destinado en compilación y pruebas, un tiempo normal para el desarrollo de la aplicación, sin embargo el porcentaje de evaluación (COQ) si es demasiado bajo (4.55 %) es decir, muy poco tiempo se destinó a revisiones de diseño y codificación.

Para los componentes de calidad PSP recomienda unos valores, por ejemplo para la densidad de defectos en la fase de pruebas lo recomendado es tener menos de 5 defectos por cada 1.000 líneas de código, la densidad de defectos obtenida en la fase de pruebas de la aplicación Banco de Tiempo fue de 3.67, una densidad de defectos muy buena.

El índice de calidad del proceso castiga mucho el hecho de que uno o dos componentes sea demasiado bajo, para empezar PSP recomienda que el tiempo destinado al diseño sea mayor al tiempo destinado en codificación, también que el tiempo en revisión de diseño sea por lo menos la mitad del tiempo destinado al diseño y lo mismo para codificación, estos componentes fueron demasiado bajos en el desarrollo de la aplicación. Con respecto a los defectos en compilación, PSP sugiere que sean menos de 10 por cada 1.000 líneas de código y para pruebas que sean menos de 5 por cada 1.000 líneas de código, en estos 2 componentes el resultado fue muy bueno.

El apalancamiento de eliminación de defectos que para la revisión de diseño fue de 2.23 y para la revisión de codificación fue de 3.6 son valores buenos.

7. Plan de mejora del proceso

7.1. Problemas del proceso

Uno de los problemas encontrados en el proceso fue el desconocimiento de Reactjs, uno de los lenguajes de programación usados en la aplicación, a pesar de que Reactjs es una librería hecha en Javascript, la forma en como se utiliza es muy diferente, esto generó que se llevara más tiempo en las fase de codificación y que aumentó también considerablemente la cantidad de defectos por la inexperiencia en el lenguaje.

7.2. Sugerencias de mejora

De los resultados obtenidos en las métricas de calidad podemos generar unas sugerencias de mejora, una de ellas es destinar más tiempo en la revisión del diseño y de la codificación, esto con el fin de ser más efectivos en la remoción de defectos y ayudar a mejorar la calidad de la aplicación [28], PSP hace mucho énfasis en hallar defectos antes de las fases de compilación y pruebas, esto implica no sólo destinar más tiempo, si no hacerlo bien, PSP sugiere hacer las revisiones en una herramienta diferente a la donde se hizo el diseño o el código, incluso recomienda imprimir el diseño o el código para hacer las revisiones.

También es importante conocer muy bien las herramientas a utilizar antes de empezar el proceso, ya que hay entornos de programación (IDE) que hacen revisión del código para detectar errores de sintaxis y que sean solucionados antes de pasar a las fases de compilación y/o pruebas.

7.3. Lecciones aprendidas

Las métricas obtenidas en el desarrollo de la aplicación nos deja algunas lecciones para el desarrollo de aplicaciones futuras, entre ellas que las prácticas utilizadas (estimación de tiempos, revisiones de código, estándares, medición de tiempos) son realmente muy útiles para medir y mejorar la calidad de una aplicación y que es importante utilizar para generar productos de calidad.

La estimación realizada durante la fase de planeación nos arrojó que el tiempo planeado en la fase de codificación fue mucho menor (25% aproximadamente) que el llevado en el desarrollo de la aplicación, también que la cantidad de defectos inyectados que se planearon fue muy bajo (84% aproximadamente) que el real, estas métricas son útiles para hacer una mejor planeación en proyectos futuros.

Las fases de revisión de diseño y revisión de codificación son fases muy importantes dentro de un proceso de desarrollo, destinar un buen tiempo a estas fases definitivamente disminuiría la cantidad de fallas de una aplicación.

VIII Bibliografía

- [1] bancodeltiempo.sanjavier.es, (30 de noviembre de 2021) Banco del tiempo / Qué es. Recuperado de: <http://bancodeltiempo.sanjavier.es/bdl.php?id=1>
- [2] data.worldbank.org, (16 de noviembre de 2021) Gini index (World Bank estimate). Recuperado de: <https://data.worldbank.org/indicator/SI.POV.GINI>
- [3] Norman H. Nie D. Sunshine Hillygus, (2002), The Impact Of Internet Use On Sociability: Time-diary Findings. It&Society
- [4] Neil Schoenherr, (Febrero 13 de 2019). Altruistic animals. Recuperado de: <https://source.wustl.edu/2009/02/altruistic-animals/>
- [5] lifeoptimizer.org, (30 de noviembre de 2021), How to Succeed in Life by Learning from Others, Recuperado de: <https://www.lifeoptimizer.org/2013/08/13/how-to-succeed-in-life/>
- [6] bdtonline.org, (30 de noviembre de 2021), Banco del tiempo | Encuentra tu banco del tiempo, Recuperado de: <http://www.bdtonline.org/>
- [7] Innovation in Aging, Volume 1, Issue suppl_1, July 2017, Page 1034, https://academic.oup.com/innovateage/article/1/suppl_1/1034/3900897
- [8] Cahn, E. (2000). No more throwaway people: The coproduction imperative. Essential Books.
- [9] IEEE. (2014). SWEBOK Guide V3.0: Guide to the Software Engineering Body of Knowledge

[10] masadelante.com, (30 de noviembre de 2021), ¿Qué significa World Wide Web, web o www? - Definición de World Wide Web, web o www. Recuperado de:

<http://www.masadelante.com/faqs/www>

[11] computer.org, (30 de noviembre de 2021), Software and Systems Engineering Vocabulary. Recuperado de: www.computer.org/sevocab

[12] Watts S. Humphrey. (2005). PSP: A Self-Improvement Process for Software Engineers, Addison Wesley

[13] timeoverflow.org, (30 de noviembre de 2021), TimeOverflow. El software diseñado por y para los Bancos de Tiempo. Recuperado de:

<https://www.timeoverflow.org/pages/about>

[14] Time Banks UK, (30 de noviembre de 2021), Welcome to Time Banks UK and the time banking movement. Recuperado de: <https://timebanking.org/>

[15] R.E.R.S., (21 de octubre de 2021), Le site des réseaux d'échanges réciproques de savoirs romand. Recuperado de: <https://www.rers.ch/>

[16] TimeBanks.Org, (1 de noviembre de 2021), Timebanks | TimeBanks.Org. Recuperado de: <https://timebanks.org/>

[17] Seyfang, G., & Smith, K. (2002). The Time of our Lives: Using Time Banking for Neighbourhood Renewal and Community Capacity Building.

[18] Parveen Mor, Rajesh Kumar Tyagi, Charu Jain, Deepak Kumar Verma, "A Systematic Review and Analysis of Blockchain Technology for Corporate Remittance and Settlement Process", Computational Intelligence and Communication Technologies (CCICT) 2021 Fourth International Conference on, pp. 121-128, 2021.

[19] python.org, (30 de noviembre de 2021). PEP 8 -- Style Guide for Python Code. Recuperado de <https://www.python.org/dev/peps/pep-0008/>

[20] es6-features.org, (20 de septiembre de 2021). ECMAScript 6 — New Features: Overview & Comparison. Recuperado de <http://es6-features.org/>

[21] djangoproject.com, (10 de octubre de 2021), Django project, the web framework for perfectionists with deadlines. Recuperado de: <https://www.djangoproject.com>

[22] techopedia.com, (10 de noviembre de 2021), HTML5. Recuperado de: <https://www.techopedia.com/definition/1891/html5>

[23] tc2-proyectodegradoingdesistemas.blogspot.com.co, (10 de noviembre de 2021). ¿Qué es y para qué sirve CSS3?. Recuperado de: <http://tc2-proyectodegradoingdesistemas.blogspot.com.co/2012/05/que-es-y-para-que-sirve-css3.html>

[24] maestrosdelweb.com, (10 de noviembre de 2021). ¿Qué es Javascript?. Recuperado de: <http://www.maestrosdelweb.com/que-es-javascript/>

[25] devcode.la, (10 de abril de 2021). ¿Cómo funciona React.js? .Recuperado de: <https://devcode.la/blog/como-funciona-reactjs/>

[26] microbuffer.wordpress.com, (10 de noviembre de 2021). ¿Qué es PostgreSQL? . Recuperado de: <https://microbuffer.wordpress.com/2011/05/04/que-es-postgresql/>

[27] about.gitlab.com, (15 de noviembre de 2021). Features. Recuperado de: <https://about.gitlab.com/features/>

[28] C. F. Kemerer and M. C. Paulk, "The Impact of Design and Code Reviews on Software Quality: An Empirical Study Based on PSP Data," in IEEE Transactions on

Software Engineering, vol. 35, no. 4, pp. 534-550, July-Aug. 2009, doi:
10.1109/TSE.2009.27.