| http://www.duoc.cl/sites/default/files/logo_summit_0.png |
| --- |
| *Capstone* |
| *Proyecto: “ChamBee”* |
|  |
| **Revisión*: [1.6]*** |
|  |

Integrantes

Arancíbia Matías

Fernández Diego

Fuentes Bayron

## Control de Versiones

| **Fecha** | **Revisión** | **Autor** | **Modificación** |
| --- | --- | --- | --- |
| *05-09-25* | *1.0* | *Matias Arancibia*  *Diego Fernandez*  *Bayron Fuentes* | *Creación de documento, objetivos y requerimientos.* |
| *15-09-25* | *1.1* | *Diego Fernandez* | *Evaluación de Problemas.* |
| *25-09-25* | *1.2* | *Bayron Fuentes* | *Factibilidades* |
| *02-10-25* | *1.3* | *Matias Arancibia* | *Analisis FODA* |
| *09-10-25* | *1.4* | *Diego Fernandez* | *Metodología de Trabajo* |
| *10-10-25* | *1.5* | *Matias Arancibia* | *Arquitectura y explicación.* |
| *12-10-25* | *1.6* | *Matias Arancibia* | *Modelo de datos* |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## 

## Índice

[**Control de Versiones 2**](#_up4e86getnnh)

[**Índice 3**](#_c3gsf8g583lx)

[**Introducción 4**](#_wh3ni96beqxy)

[**Situación Actual 4**](#_a08ar26cbxc2)

[**Analisis FODA 5**](#_2h39zevox1xy)

[**Evaluación del Problema 6**](#_i8cynpwr9saf)

[**Requerimientos 6**](#_hnem8wgjzeg4)

[**Restricciones 7**](#_szatciodpryv)

[**Factibilidad Legal 7**](#_4wowfbrvkhgm)

[**Factibilidad Técnica 7**](#_9x5n6axine7s)

[**Factibilidad Operacional 8**](#_pchzy6felom9)

[**Metodología de trabajo 8**](#_mjikjixx3dgu)

[1. Planificación del Sprint 9](#_mldq2x673ud0)

[2. Ejecución del Sprint 9](#_dvrz1fxvpzw9)

[3. Revisión del Sprint 9](#_6pyzimj4dcv0)

[**Arquitectura 10**](#_jgcyjs6gqq0w)

[**Modelo de Datos 11**](#_1q5wpsmvo3yo)

## Introducción

El proyecto "ChamBee" consiste en una plataforma digital diseñada para conectar a prestadores de oficios con vecinos de la comuna. Su objetivo general es desarrollar una plataforma que gestione y potencie los oficios a nivel comunal, asegurando eficiencia, confianza y calidad del servicio. La plataforma busca resolver una necesidad social y tecnológica, fomentando la empleabilidad local y la digitalización de servicios comunales, con el respaldo de la municipalidad.

## Situación Actual

La situación actual se caracteriza por varios desafíos, particularmente en comunas alejadas del centro:

* Alta demanda de servicios en comunas que no son el centro urbano.
* Falta de empleo local.
* Baja digitalización en las bolsas laborales municipales existentes.
* Necesidad de generar seguridad y confianza al momento de contratar prestadores de servicios.

## 

## Analisis FODA

| Aspecto | Detalle |
| --- | --- |
| Fortalezas (F) | Plataforma digital con respaldo municipal. Inclusión de beneficiarios de programas de capacitación. Permite gestión completa de perfil de prestador (documentos, portafolio). Sistema de reputación y feedback (puntaje 1 a 5 estrellas y comentarios). Chatbot para guía y filtrado específico de servicios. |
| Oportunidades (O) | Alta demanda de servicios en comunas alejadas del centro de la ciudad. Integración con programas locales de empleo y capacitación. Modernización de la gestión municipal. Posibilidad de Fase 2 (Contratos, Automatización de certificaciones, Alianzas con otras Apps). |
| Debilidades (D) | Dependencia de alianzas estratégicas con municipalidades para la administración y validación. El éxito depende de la adopción y registro activo de prestadores y clientes. |
| Amenazas (A) | Necesidad de asegurar la seguridad y privacidad de la comunicación y datos. Riesgo de errores en el cálculo de la nota promedio o en la validación de reseñas. CAmbio de administración municipal cada 4 años |

## 

## 

## 

## 

## Evaluación del Problema

El problema principal radica en la desconexión entre la alta demanda de servicios en las comunas y la disponibilidad de prestadores de oficio locales , lo que se agrava por la baja digitalización de los sistemas de empleo municipales. Adicionalmente, existe una brecha de seguridad y confianza en la contratación, lo que dificulta a los vecinos acceder a servicios de calidad y a los prestadores locales obtener trabajo.

El problema tiene un impacto directo en el empleo local y la calidad de vida comunal. La falta de una plataforma centralizada y validada obstaculiza:

* El acceso a oportunidades laborales para prestadores locales.
* La modernización de la gestión municipal.
* La capacidad de dar respuesta rápida a servicios (emergencia y programados).
* El fortalecimiento de la confianza ciudadana en los servicios contratados.

La propuesta de "ChamBee" busca solucionar esto mediante la administración y validación de prestadores a cargo de la municipalidad , generando confianza y potenciando a los beneficiarios de programas de capacitación y empleo.

## Requerimientos

* Gestión de Perfil de Prestador: Registro de usuarios, perfil con documentos de verificación y portafolio de trabajos, agenda tipo calendario, y posibilidad de ofrecer más de un servicio.
* Búsqueda y Conexión: Búsqueda con filtros (servicio, ubicación) , conexión con API de Chatbot para filtro específico , y notificaciones en tiempo real.
* Validación Municipal: Panel de control para el Administrador Municipal (aprobación/rechazo de prestadores).
* Reputación y Feedback: Encuesta de satisfacción , evaluación con puntaje de 1 a 5 estrellas y comentario , y cálculo de nota promedio.
* Comunicación: Chat uno a uno entre Cliente y Prestador post-Match.
* Cotización: Módulo para que el prestador genere, envíe y edite cotizaciones detalladas (materiales, mano de obra, costos).

## Restricciones

* Tecnológicas Propuestas: Se sugiere utilizar React para el Frontend, Python para el Backend , SQL server como Base de Datos , y GC (Google Cloud) para servicios.
* Seguridad y Privacidad: La comunicación debe ser privada (solo un prestador y un cliente) y solo accesible a usuarios autenticados.

## Factibilidad Legal

La alianza estratégica con municipalidades para la administración y validación de prestadores y la integración con programas locales de empleo sugiere que el proyecto buscará operar dentro del marco legal de colaboración público-privada o bajo un modelo de apoyo al empleo comunal, requiriendo acuerdos formales. El manejo de información de prestadores y clientes (documentos de verificación, información personal) implica la necesidad de cumplir con la legislación de protección de datos y privacidad.

## Factibilidad Técnica

El proyecto propone el uso de tecnologías modernas y robustas (React, Python, SQL server, GCP), que son adecuadas para desarrollar una plataforma web con las funcionalidades requeridas:

* Desarrollo de Backend/Frontend: Es factible construir el sistema de perfiles, búsqueda, reputación, y cotizaciones.
* Chat en Tiempo Real: Se planea la implementación de servicios de mensajería usando WebSocket o un Servicio GCP, lo cual es técnicamente viable para la comunicación instantánea.
* Conexión con Chatbot: La integración con una API de chatbot es técnicamente factible.
* Almacenamiento: SQL server permite la gestión de los datos de usuarios, servicios, reseñas, y el histórico de mensajes.

## Factibilidad Operacional

El proyecto está diseñado para ser operacionalmente viable a través de:

* Roles Claros: Distinción entre Administrador (Municipalidad) , Cliente , y Prestador , lo que facilita la gestión del flujo de trabajo.
* Validación de Confianza: La Municipalidad asume el rol de validador , lo que asegura la calidad y fiabilidad de los prestadores antes de su registro (Aprobación/Rechazo).
* Mejora Continua: El sistema incluye funcionalidades de evaluación y feedback, lo que permite mantener la calidad del servicio a largo plazo.
* Beneficios Comunitarios: Al enfocarse en resolver la alta demanda de servicios y potenciar el empleo local, el proyecto se integra positivamente en las operaciones comunales.

## Metodología de trabajo

Para abordar el desarrollo del proyecto "ChamBee" (Plataforma de Oficios), se utilizará una metodología ágil, específicamente Scrum. Esta metodología es ideal para proyectos de software con requisitos que pueden evolucionar, ya que promueve la flexibilidad y la colaboración constante.

El objetivo de aplicar Scrum es asegurar un desarrollo flexible, la incorporación oportuna de feedback y un producto final útil y exitoso.

**Fases de Implementación bajo Scrum**

La implementación del proyecto se dividirá en ciclos de trabajo denominados Sprints, con las siguientes fases:

#### 1. Planificación del Sprint

* Descripción: Al inicio de cada ciclo, se define qué funcionalidades se desarrollarán en ese período.
* Proceso: El equipo de desarrollo, el Product Owner, y el Scrum Master seleccionan y mueven tareas del Product Backlog (lista priorizada de funcionalidades) al Sprint Backlog.
* Resultado: Las funcionalidades se desglosan en tareas más pequeñas y estimables, y el equipo se compromete a entregar un "incremento de producto" funcional al final del sprint.

#### 2. Ejecución del Sprint

* Duración: El equipo trabajará en las tareas seleccionadas durante un período fijo, usualmente de 2 a 4 semanas.
* Método de Trabajo:
  + Daily Scrum (Reunión Diaria): Una reunión diaria de 15 minutos donde cada miembro comparte el progreso, las tareas del día, y los obstáculos. Esto fomenta la transparencia y la resolución rápida de problemas.
  + Colaboración Continua: El equipo mantendrá una comunicación constante y resolverá dudas a medida que surjan.

#### 3. Revisión del Sprint

* Descripción: Al finalizar el sprint, se realiza una reunión para mostrar el incremento de producto funcional a los stakeholders.
* Proceso: Se presentan las funcionalidades completadas y se recoge feedback. Este feedback es crucial para adaptar el Product Backlog y alinear el proyecto con las necesidades.

## Arquitectura

## 

Para la construcción del diagrama de arquitectura, se utilizaron los siguientes servicios de GCP en conjunto con tecnologías como:

Cloud Storage

* Un servicio de almacenamiento de objetos altamente escalable y duradero. Se utilizará en este caso para almacenar los datos en formato Parquet para el procesamiento batch.

Cloud SQL

* Servicio de base de datos relacional completamente administrado, diseñado para facilitar la gestión de bases de datos de SQL Server.

Cloud Engine

* Servicio que proporciona máquinas virtuales escalables y seguras para alquilar potencia informática en la nube, sin necesidad de gestionar servidores físicos.

Cloud IAM

* Un servicio para gestionar el acceso a los recursos de GCP, permitiendo controlar quién tiene permiso para hacer qué.

Key Management Service (KMS)

* Un servicio para gestionar claves de cifrado de forma segura. Se utiliza para cifrar datos en reposo y en tránsito.

Logging

* Un servicio para recopilar, almacenar y analizar registros de todos los servicios de GCP. Es fundamental para la monitorización, la depuración y la auditoría.

Cloud Armor

* Un servicio de seguridad de aplicaciones web que protege las aplicaciones de ataques DDoS y otras amenazas.

React

* Biblioteca de JavaScript para crear interfaces de usuario interactivas y eficientes para aplicaciones web y móviles. Permite construir estas interfaces a partir de componentes reutilizables, que son piezas de código que combinan estructura HTML y lógica de JavaScript.

FastAPI

* Framework web moderno y de alto rendimiento para crear APIs con Python. Busca lograr velocidad y validación automática de datos. Su diseño facilita el desarrollo de APIs eficientes, rápidas y escalables.

## Modelo de Datos

Para el presente proyecto se utilizará el lenguaje SQL para bases de datos relacionales, usando el motor SQL Server en el servicio de GCP Cloud SQL. Con el objetivo de satisfacer todas las necesidades del proyecto relacionado al manejo de datos, se genera el siguiente esquema.

* [Esquema de datos](https://drive.google.com/file/d/1C_cZxNkwjsRmVuQU2z8jVjUxZlC73Y-H/view?usp=drive_link)