# 

**Iniciando el Desarrollo de Software  
MatchWork**

***Nombre Estudiante:*** *Sebastián Santibáñez, Dario Olivares*

***Asignatura****: Taller Aplicado de Software*

***Carrera****: Ingeniería en Desarrollo de Software*

***Profesor****: Carlos Valverde*

***Fecha****: 19/06/2025*

Contenido

[1. Introducción 3](#_Toc203143917)

[2. Resumen Ejecutivo 4](#_Toc203143918)

[3. Links con Información del Proyecto 5](#_Toc203143919)

[4. Link con Acceso al Sistema 6](#_Toc203143920)

[5. Desarrollo del Sistema 7](#_Toc203143921)

[6. Lenguajes de Programación Utilizados 8](#_Toc203143922)

[7. Dificultades de esta Etapa 9](#_Toc203143923)

## 1. Introducción

El presente documento corresponde a la entrega de la semana 5 del proyecto MatchWork, una plataforma web de conexión laboral que facilita el encuentro entre profesionales y empresas mediante un sistema de emparejamiento inteligente basado en habilidades, comunicación en tiempo real y videollamadas integradas. En esta fase, se detallan los avances realizados en el desarrollo del sistema, incluyendo la implementación de funcionalidades clave como la autenticación, el perfil profesional, la publicación de empleos, el sistema de postulaciones, el chat y la videollamada.

## 2. Resumen Ejecutivo

Durante esta etapa se logró avanzar significativamente en el desarrollo de MatchWork, consolidando las bases técnicas del sistema y conectando sus principales módulos. Se implementó una arquitectura de microservicios con Spring Boot desplegada en instancias EC2 de AWS, utilizando Oracle como base de datos principal y MongoDB Atlas para el sistema de mensajería. La autenticación se resolvió mediante AWS Cognito, y los archivos de usuario como fotos de perfil y currículums se almacenan en AWS S3. Ya se encuentran operativas funcionalidades críticas como el registro y login de usuarios, creación y edición de perfiles, publicación de ofertas laborales, postulación con validación, sistema de match automático por habilidades, chat en tiempo real y videollamadas directas integradas con Jitsi Meet.

Estos avances permiten validar el correcto funcionamiento de la lógica principal del sistema, sentando las bases para comenzar el proceso de pruebas, ajustes y optimización de cara a las entregas finales.

## 3. Links con Información del Proyecto

El proyecto MatchWork está dividido en múltiples repositorios públicos en GitHub, cada uno correspondiente a un componente del sistema basado en microservicios.

Frontend Angular (desplegado en Vercel):

* https://github.com/DarioOlivares2001/matchwork

Backend Spring Boot:

* API Gateway: <https://github.com/DarioOlivares2001/apigateway>
* Servicio de Chat: <https://github.com/DarioOlivares2001/micro-matchworkchat>
* Servicio de Usuarios: <https://github.com/DarioOlivares2001/user-services>
* Servicio de Trabajos: <https://github.com/DarioOlivares2001/jobservice>

Cada repositorio contiene el código fuente completo, estructuras organizadas por capas, y secciones de documentación básica. Los archivos README serán incluidos próximamente para facilitar la comprensión del código y el despliegue del sistema.

## 4. Link con Acceso al Sistema

MatchWork ya se encuentra desplegado en línea, accesible desde:

* Frontend Angular (SSR con Vercel):

<https://matchwork-wo14.vercel.app/>

Microservicios Backend desplegados en EC2 – AWS (t2.micro, zona us-east-1):

|  |  |  |  |
| --- | --- | --- | --- |
| Microservicios | IP Publica | Puerto | Función Principal |
| API Gateway | 52.207.17.209 | 8080 | Punto de entrada unificado (CORS, JWT, rutas) |
| Chat Service | 34.193.228.160 | 8080 | WebSocket + MongoDB (mensajes en tiempo real) |
| Job Service | 35.174.51.1 | 8080 | CRUD de empleos y matching por habilidades |
| User Service | 54.90.98.100 | 8080 | Registro, login y perfil de usuarios |

Los servicios son accesibles únicamente a través del API Gateway para mantener el flujo seguro y controlado. Además, el sistema utiliza:

* Autenticación con AWS Cognito (usuario, login, roles)
* Almacenamiento de imágenes, currículums y logos en AWS S3
* Mensajes en tiempo real con WebSocket + MongoDB Atlas
* Videollamadas integradas con Jitsi Meet desde el frontend

## 5. Desarrollo del Sistema

Actualmente, el sistema MatchWork se encuentra funcional y desplegado en producción. Los módulos implementados y completamente operativos son:

* Autenticación y registro: Integrado con AWS Cognito desde el frontend.
* Perfiles de usuario: Empresas y profesionales pueden editar su información, subir imágenes y currículum.
* Publicación de trabajos: Las empresas pueden publicar, editar y eliminar ofertas.
* Postulación y matching: Profesionales pueden postular y reciben sugerencias según sus habilidades.
* Chat en tiempo real: Implementado con WebSocket y MongoDB.
* Videollamadas: Funcionalidad activa mediante integración con Jitsi Meet.
* Dashboard de usuarios: Interfaz diferenciada según el tipo de usuario.

No hay funcionalidades pendientes; el sistema está listo para evaluación final.

## 6. Lenguajes de Programación Utilizados

Frontend:

* Angular 18 (standalone + SSR)
* TypeScript
* Bootstrap 5

Backend:

* Java 17 (Spring Boot 3)
* Spring Cloud Gateway
* Spring Data JPA

Infraestructura:

* Docker
* AWS EC2 (para microservicios)
* AWS S3 (para avatar, foto perfil y curriculum)
* AWS Cognito (para credenciales)

Base de datos:

* Oracle Cloud Database (para perfiles, y trabajos)
* MongoDB Atlas (para chat)

## 7. Dificultades de esta Etapa

* Configuración de CORS y validación de tokens JWT en el API Gateway para permitir comunicación entre Vercel y los microservicios.
* Integración con AWS Cognito desde Angular fue compleja por los flujos de confirmación de cuenta y recuperación de contraseña.
* Almacenamiento en AWS S3 requirió configuración de roles y permisos adecuados para acceder a los archivos desde Angular.
* Despliegue en EC2 y configuración de puertos, seguridad y dominios fue una etapa crítica que requirió varios ajustes.