Informe Final ATP

Software - IntegraJob

Chatbot de búsqueda laboral para personas con neuro divergencia y movilidad reducida

Capstone | Proyecto de título

Duoc UC | Sede Padre Alonso de Ovalle

Profesor: Felix Cifuentes

Integrantes:

Carlos Colmenarez | Rut: 27.009.196-2

Francisco Eljuri |

Javier Rivas | Rut: 19.512.852-9

14/11/2025

Índice

[**Abstract 3**](#_fgi0e45t3l5w)

[English 3](#_8kuns38cyfid)

[**Relevancia del proyecto APT 5**](#_ggtyv6u094jm)

[Problema y contexto: 5](#_lhqgfhkcsw5c)

[**Objetivos 6**](#_5m26ii2gle1w)

[General: 6](#_d5ude4y2mmpa)

[Objetivos específicos: 6](#_9qkqbje53764)

[**Metodología 7**](#_jt6sexggdobc)

[Metodología aplicada: 7](#_u8xuyu8tx5vt)

[Justificación de pertinencia para cumplir los objetivos: 7](#_voqdoz1fqskv)

[Distribución del trabajo y Roles: 7](#_jqvk4ylku09t)

[**Desarrollo del proyecto 9**](#_dhi1oog9o6ib)

[Etapas / Actividades realizadas: 9](#_z9b1w6dbusdt)

[1. Pruebas unitarias y de integración 11](#_4q1cg6ivcfg7)

[2. Verificación de accesibilidad conversacional 12](#_p20ulym8778e)

[3. Registro y trazabilidad de hallazgos 12](#_kiic9hsgao9l)

[**Entregables 13**](#_mlui4049q14x)

[**Evidencias en Git / gestión 13**](#_qm5r7gckg8m0)

[**Criterios de salida 14**](#_ssjrbeuqberb)

[Trazabilidad: 17](#_r2k1ldejsgyq)

[Facilitadores y dificultades: 17](#_e9azeleukdxj)

[Ajustes y cómo se abordaron: 18](#_b7dc6e8pobqa)

[**Evidencias 19**](#_fnfm86nrokju)

[**Resultados clave del MVP: 26**](#_bdee4m6hwn5f)

[Métricas:   
  
Rendimiento 27](#_cznyuaye2o0r)

[Hallazgos clave: 28](#_favenn3rp9yn)

[**Lecciones aprendidas técnicas y de gestión:  
  
Técnicas 29**](#_837hr8vlq3i1)

[Gestión y trabajo en equipo 29](#_chielnyvwmte)

[**Intereses y proyecciones profesionales (Individual): 31**](#_jp4cxon1m3ve)

[Carlos Colmenarez 31](#_4424bnwhft5g)

[Aporte del proyecto a tus intereses: 31](#_7pvim83uf5j3)

[Proyecciones laborales (post APT): 31](#_f5xcpopkkdmg)

[Javier Rivas 31](#_fkpqclramdqd)

[Aporte del proyecto a tus intereses 31](#_y42nwl7wpg6y)

[Proyecciones laborales (post APT) 31](#_t8w5brmt0wdv)

[Francisco Eljuri 32](#_k4nv2ihik69d)

[Aporte del proyecto a tus intereses: 32](#_lww7a5u6s5wz)

[Proyecciones laborales (post APT): 32](#_m5v6122gbjcx)

[**Conclusiones individuales 33**](#_7evf9vwq3oif)

[Carlos Colmenarez 33](#_pqajmb1p4q4a)

[Inglés 33](#_jl46lktobcuq)

[Javier Rivas 34](#_9uwk5h3hw3dh)

[Inglés 34](#_1elm301xod5m)

[Francisco Eljuri 35](#_ckuw4edzbmmq)

[Inglés 35](#_i6rbwks6bnnj)

[**Reflexiones 36**](#_s27t96gl182v)

[Español 36](#_89ra2ye3133b)

## **Abstract**

IntegraJob es una plataforma inclusiva de intermediación laboral que centraliza ofertas desde fuentes externas, principalmente la API de la Bolsa Nacional de Empleo (BNE) y las adapta para personas con necesidades de accesibilidad y neurodivergencias, reduciendo barreras de entrada al empleo formal. La solución se compone de un Scraping elaborado con Python para integración y normalización de datos, un backend Django con Postgresql para la gestión de entidades y una interfaz web React orientada a usabilidad y cumplimiento de estándares WCAG. La arquitectura modular basada en microservicios contenedorizados (Docker) prioriza interoperabilidad, escalabilidad horizontal y seguridad (OAuth2, control de acceso por roles), asegurando flujos robustos para consulta y recomendación de vacantes. En el semestre se ejecutó un enfoque ágil con hitos de integración BNE (OAuth2, consumo de endpoints activos), normalización taxonómica y persistencia, más pruebas unitarias e integración para validar calidad y trazabilidad. Como evidencias, el proyecto mantiene documentación técnica, código y artefactos de QA versionados en Git, cumpliendo la estructura exigida por la asignatura. Los resultados del MVP incluyen la ingesta confiable de ofertas, clasificación por criterios de accesibilidad y búsqueda filtrada por ubicación, preparando la base para funcionalidades avanzadas a futuro. Entre las proyecciones destacan interacción por voz, soporte multilingüe y algoritmos de IA para mejorar el “match” candidato-puesto. En conjunto, IntegraJob evidencia competencias de desarrollo e integración de software, modelado de datos, pruebas y gestión de proyectos, y entrega un aporte tangible al ecosistema de empleo inclusivo.

### English

IntegraJob is an inclusive job placement platform that centralizes job offers from external sources, mainly the National Employment Exchange (BNE) API, and adapts them for people with accessibility needs and neurodiversities, reducing barriers to formal employment. The solution consists of Python scraping for data integration and normalization, a Django backend with Postgresql for entity management, and a React web interface focused on usability and compliance with WCAG standards. The modular architecture based on containerized microservices (Docker) prioritizes interoperability, horizontal scalability, and security (OAuth2, role-based access control), ensuring robust flows for job vacancy queries and recommendations. During the semester, an agile approach was implemented with milestones for BNE integration (OAuth2, active endpoint consumption), taxonomic normalization, and persistence, plus unit and integration testing to validate quality and traceability. As evidence, the project maintains technical documentation, code, and QA artifacts versioned in Git, complying with the structure required by the course. The MVP results include reliable intake of offers, classification by accessibility criteria, and filtered search by location, laying the foundation for advanced features in the future. Notable projections include voice interaction, multilingual support, and AI algorithms to improve candidate-job matching. Overall, IntegraJob demonstrates competencies in software development and integration, data modeling, testing, and project management, and makes a tangible contribution to the inclusive employment ecosystem.

## **Relevancia del proyecto APT**

### Problema y contexto:

El proyecto aborda la baja accesibilidad y fragmentación de ofertas laborales relevantes para personas con requerimientos de accesibilidad y neurodivergencias en Chile. Esto impacta su empleabilidad y permanencia en el mercado formal. Es relevante para la profesión porque demanda integración de datos (ETL), diseño de modelos relacionales, servicios API seguros, accesibilidad web, prácticas de QA/DevOps, y de desarrollo (Backend/Frontend); y para el campo laboral porque habilita procesos de búsqueda y recomendación confiables, auditables y escalables.

**Ubicación:**

Chile, foco en la ciudad de Santiago y contexto académico APT (Escuela de Informática, 2025).

Marco normativo y fuentes locales: Ley 21.015 (inclusión laboral), Ley 19.628 (protección de datos), y la Bolsa Nacional de Empleo (BNE) como principal proveedor de ofertas, además de extracción de ofertas de laborum y computrabajo.

**Población afectada:**

Personas en búsqueda de empleo con necesidades de accesibilidad y/o neurodivergencias; además, empresas con interés en contratación inclusiva; y actores del ecosistema laboral (intermediadores, servicios públicos) que requieren información normalizada y trazable.

**Aporte de valor:**

Valor real en el MVP: consolidación de ofertas desde BNE, normalización taxonómica, búsqueda por filtros (ubicación, accesibilidad) y base para recomendaciones. Aporta trazabilidad, cumplimiento normativo y reducción de barreras de acceso al empleo.

## **Objetivos**

### General:

Diseñar, implementar y validar un MVP de plataforma inclusiva de intermediación laboral (IntegraJob) que integre ofertas de la BNE y Scraping de portales laborales (Computrabajo, Laborum), normalice y publique la información con criterios de accesibilidad, y ofrezca búsqueda/filtrado además recomendaciones básicas, sobre una arquitectura modular (Scraping + API + frontend + Backend), con seguridad y trazabilidad, demostrando competencias de desarrollo, datos, QA y gestión.

### Objetivos específicos:

**OjE1:** Integración de datos: Consumir la API BNE y normalizar ofertas con trazabilidad de ingesta.

**OjE2**: Modelo y persistencia: Diseñar e implementar el modelo relacional (ofertas, empresas, ubicaciones, etiquetas de accesibilidad, usuarios/roles).

**OjE3**: Servicios (API), Exponer endpoints de consulta y filtrado con autenticación/autorización y control de roles.

**OjE4**: Interfaz accesible: Implementar un frontend alineado a WCAG, priorizando usabilidad y claridad de requisitos/beneficios.

**OjE5**: Recomendación inicial de vacantes según preferencias del usuario.

**OjE6**: Calidad (QA). Diseñar y ejecutar pruebas unitarias e integración con criterios de aceptación y reporte de resultados.

**OjE7:** DevOps/Despliegue, contenerizar servicios (Docker) y documentar flujo de despliegue; versionar en Git con evidencias por integrante.

**OjE8**: Cumplimiento normativo: Implementar resguardo de datos personales, HTTPS y políticas de sesión.

**OjE9**: Evidencias y presentación, Entregar ERS, DAS, código, datos de prueba, actas/tablero y presentación final.

## **Metodología**

### Metodología aplicada:

Este proyecto se ha desarrollado bajo una metodología en cascada, avanzando de manera secuencial por etapas claramente definidas que aseguran control y coherencia en cada fase. Comenzamos con:

1. Inicio (Análisis): Levantamiento de requerimientos y diseño de la base de datos.
2. Backend: Desarrollo en Django con API para servicios del sistema.
3. Frontend: Interfaz en React/Vite con chatbot como punto de interacción.
4. Módulo analítico: Django/Python para procesar y clasificar ofertas laborales.
5. Scraper inclusivo: Extracción desde portales (Computrabajo, Laborum) con análisis de texto para detectar accesibilidad y transporte.
6. Validación por etapas: Cada fase se verifica antes de avanzar a la siguiente.
7. Trazabilidad y calidad: Documentación en GitHub, revisión semanal y control de progreso para asegurar calidad del producto final.

### Justificación de pertinencia para cumplir los objetivos:

El método de cascada es pertinente pues el proyecto posee requisitos base estables, entregables obligatorios secuenciales (ERS/DAS/QA/Presentación) y una ventana temporal acotada (un semestre). El enfoque reduce ambigüedad, facilita la revisión por hitos y garantiza documentación robusta para la evaluación, manteniendo trazabilidad completa cumpliendo con los requisitos, diseños y pruebas para este proyecto.

### Distribución del trabajo y Roles:

Este trabajo se organizó y se distribuyó mediante continuas reuniones en las cuales se asignaron roles y se realizaron continuas revisiones y avances, la la organización fue la siguiente:

**Roles:**   
• Líder de proyecto (Carlos Colmenarez ): planificación/alineación con docente, control de alcance, consolidación de evidencias.   
• Arquitecto/líder técnico (Francisco Eljuri/ Javier Rivas): decisiones de diseño, datos/API/seguridad y revisión de código.   
• Desarrollador ETL/Backend(Javier Rivas): ingestión y servicios.   
• Desarrollador Frontend/Accesibilidad(Francisco Eljuri): UI alineada a WCAG.   
•QA/DevOps(Francisco Eljuri / Carlos Colmenarez): plan y ejecución de pruebas, empaquetado y guía de despliegue.

**Reuniones:**   
• Semanal de estatus (30–45 min) con revisión de hitos por fase y bloqueos.   
• Gate de fase (aprobación formal de ERS, DAS, implementación y QA).   
• Actas breves con acuerdos, responsables y fechas.

**Control de avances y calidad:**   
• Git: ramas por feature, PRs con revisión, issues ligadas a requisitos/objetivos.   
• Tablero de tareas y checklist por fase.   
• Métricas del MVP: éxito de ingesta, latencia de consulta, validaciones de accesibilidad, disponibilidad en pruebas.

## 

## 

## **Desarrollo del proyecto**

### Etapas / Actividades realizadas:

La ejecución siguió la metodología Cascada y la carta Gantt, estableciendo puntos de control al cierre de cada fase. A continuación, se documentan objetivos específicos por fase, actividades, entregables, evidencias en Git y criterios de salida.

**Fase 1 – Análisis y Requisitos**

Objetivo: Solicitar y formalizar el alcance del MVP y sus restricciones, asegurando pertinencia con el APT.

Actividades clave:

* Caracterización del problema y usuarios (enfoque inclusivo y normativo).
* Levantamiento de requisitos funcionales: ingesta BNE, Scraping, normalización, filtros y recomendación inicial.
* Definición de no funcionales: seguridad (autorización, sesiones), rendimiento, accesibilidad (alineación WCAG).

Entregables:

* ERS v1.1 con criterios de aceptación y supuestos/restricciones.
* Matriz de requisitos con identificación unívoca y prioridad.
* Carta Gantt.

Evidencias en Git / gestión:

* Repositorio Git: carpeta /docs/ERS vinculadas a requisitos.

Criterios de salida : Requisitos completos y coherentes; riesgos iniciales identificados; aprobación formal del equipo.

**Fase 2 – Diseño**

Objetivo: Traducir requisitos a una solución técnica consistente y verificable.

Actividades clave:

* Modelado relacional (ofertas, empresas, ubicaciones, etiquetas de accesibilidad, usuarios/roles).
* Diseño de API (endpoints, contratos, paginación, control de errores).
* Definición de políticas de seguridad ( autenticación/autorización).
* Lineamientos de accesibilidad en UI (componentes críticos, navegación por teclado, textos alternativos).

Entregables:

* DAS (diagramas de arquitectura y de datos; API).
* Especificación de ambientes y políticas de respaldo.
* Modelado de datos.
* Documentación de Endpoint a consumir de la BNE.

Evidencias en Git / gestión:

* Repositorio Git: /docs/DAS, diagramas versionados.
* Diagrama de arquitectura.

**Fase 3 – Implementación (MVP)**

Objetivo: Construir los componentes mínimos viables respetando diseño y restricciones.

Actividades clave:

* ETL para BNE (consumo de endpoints, normalización taxonómica y trazabilidad de ingesta).
* Backend de servicios (consulta/filtrado por accesibilidad y ubicación, manejo de errores).
* Frontend accesible (UI alineada a WCAG) y contenerización (Docker).

Entregables:

* Código fuente versionado, imágenes/artefactos de despliegue.
* Guía técnica de ejecución.

Evidencias en Git / gestión:

* Ramas por feature, PRs revisadas, issues cerradas con referencia a requisitos.
* Builds reproducibles y scripts de arranque.

Criterios de salida: Cobertura funcional del MVP, ejecución reproducible en entorno de pruebas.

**Fase 4 – Pruebas y Verificación (QA)**

Objetivo:

Verificar que el MVP cumple los criterios de aceptación definidos para el motor conversacional, el motor de matching, la ingesta de datos y la API, así como los requisitos no funcionales asociados a rendimiento, accesibilidad, robustez y estabilidad.

### **1. Pruebas unitarias y de integración**

Se diseñó y ejecutó una suite de pruebas automatizadas que abarcó los componentes críticos del sistema:

* **Chatbot**:  
   Pruebas del flujo conversacional inicial, manejo de prompts ambiguos, cambios de criterios y persistencia del estado de la conversación.
* **Parser semántico** (parse\_prompt, parse\_simple\_response):  
   Validación de normalización lingüística, reconocimiento de modalidad, industria, área, transporte y salario.
* **Motor de matching** (decide\_jobs):  
   Verificación del filtrado por modalidad, área/industria, coincidencias estrictas, paginación y relajación progresiva.
* **Modelos y creación de ofertas de empleo** (JobPosting):  
   Prueba de creación, deduplicación y validación de campos requeridos.
* **API REST** (endpoints /api/chat/..., /api/jobpostings/, /api/taxonomy, /api/search):  
   Verificación de respuestas, códigos HTTP e integridad del JSON.

Todas estas pruebas se ejecutaron correctamente usando pytest/unittest y la base de datos temporal generada por Django para el entorno de testing.

### **2. Verificación de accesibilidad conversacional**

Se validaron aspectos esenciales que aportan accesibilidad:

* Mensajes del chatbot formulados en lenguaje claro y directo.
* Manejo de errores con sugerencias guiadas.
* Compatibilidad con lectura secuencial (lector de pantalla).
* Rutas consistentes y predecibles para navegación por teclado (frontend, si aplica).

### **3. Registro y trazabilidad de hallazgos**

* Se documentaron fallas de rutas, campos no coincidentes, ajustes en los tests y normalización del modelo.
* Cada corrección quedó referenciada a su endpoint o función interna del sistema.
* Se comprobaron nuevamente todos los flujos después de cada corrección.
* La suite de pruebas finalizó con **100% de éxito** (0 fallos, 0 errores).

## **Entregables**

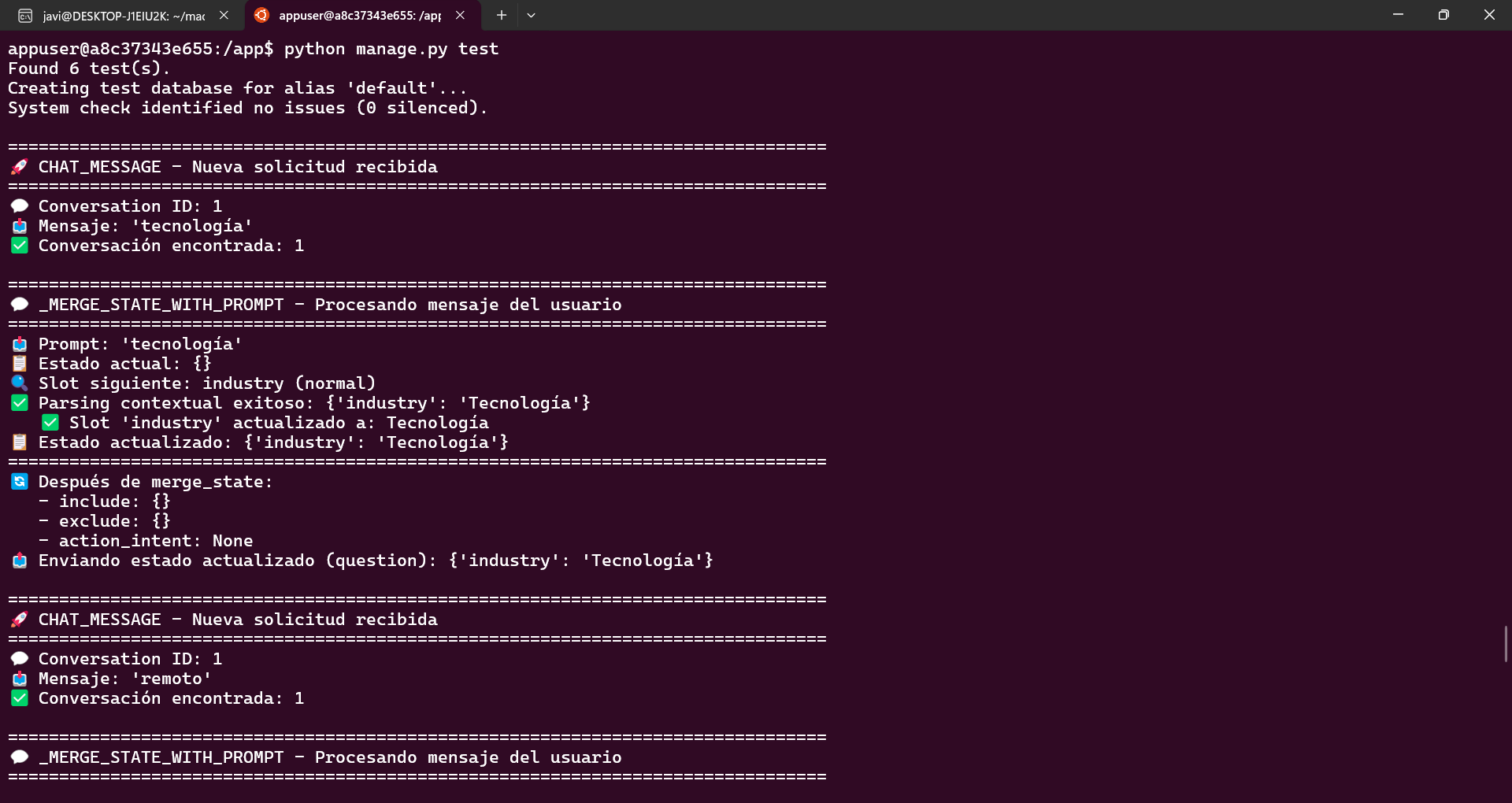
* **Plan de QA** con enfoque en componentes críticos del MVP.
* **Matriz de casos de prueba** (conversación, matching, parser, API, creación de ofertas).
* **Reportes de ejecución** generados por Django durante los tests.
* **Evidencias de corrección** en el repositorio (commits y PRs).
* **Registro de mejoras** en el parser, matching y vistas del chatbot.

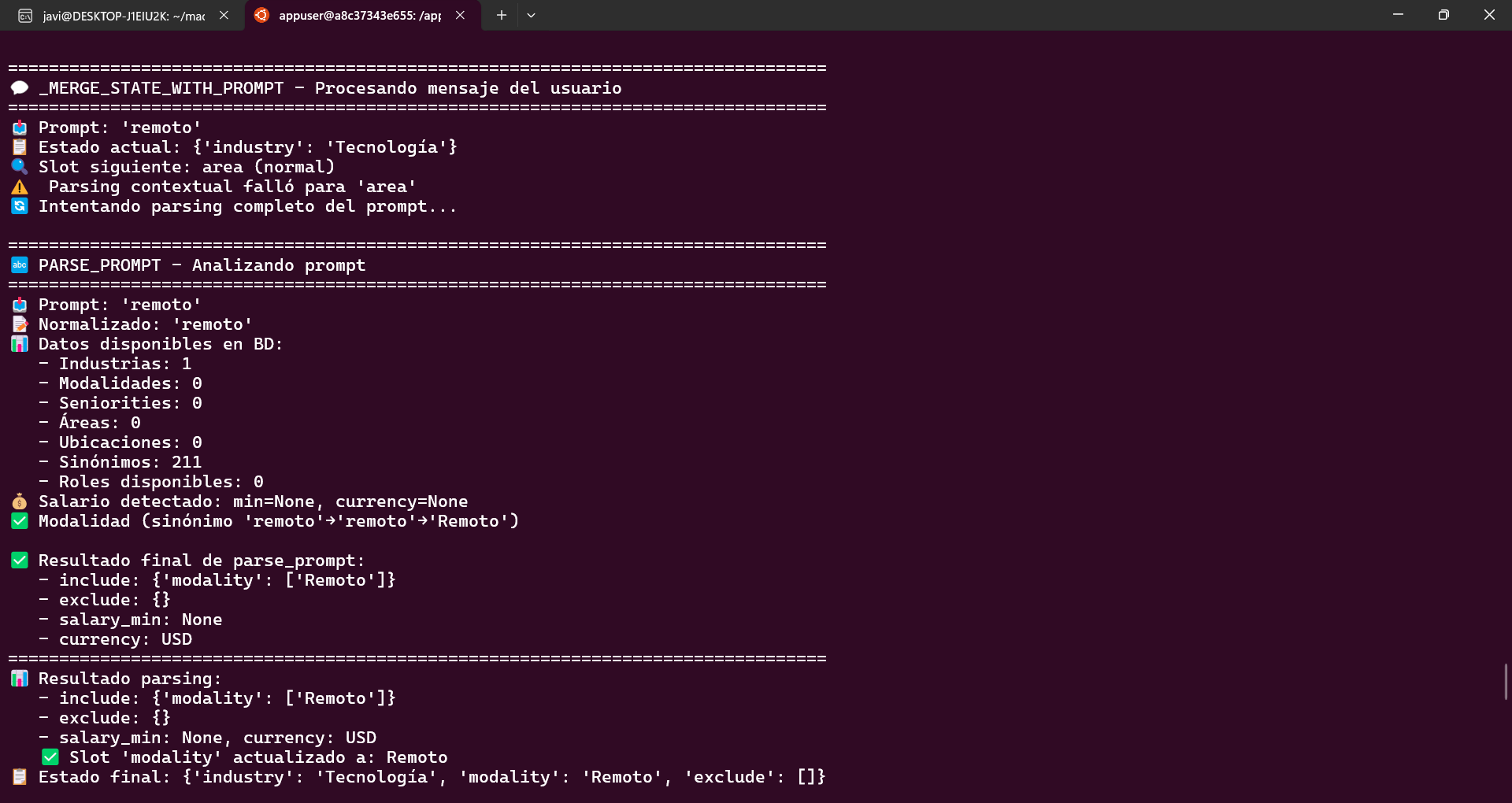
## **Evidencias en Git / gestión**

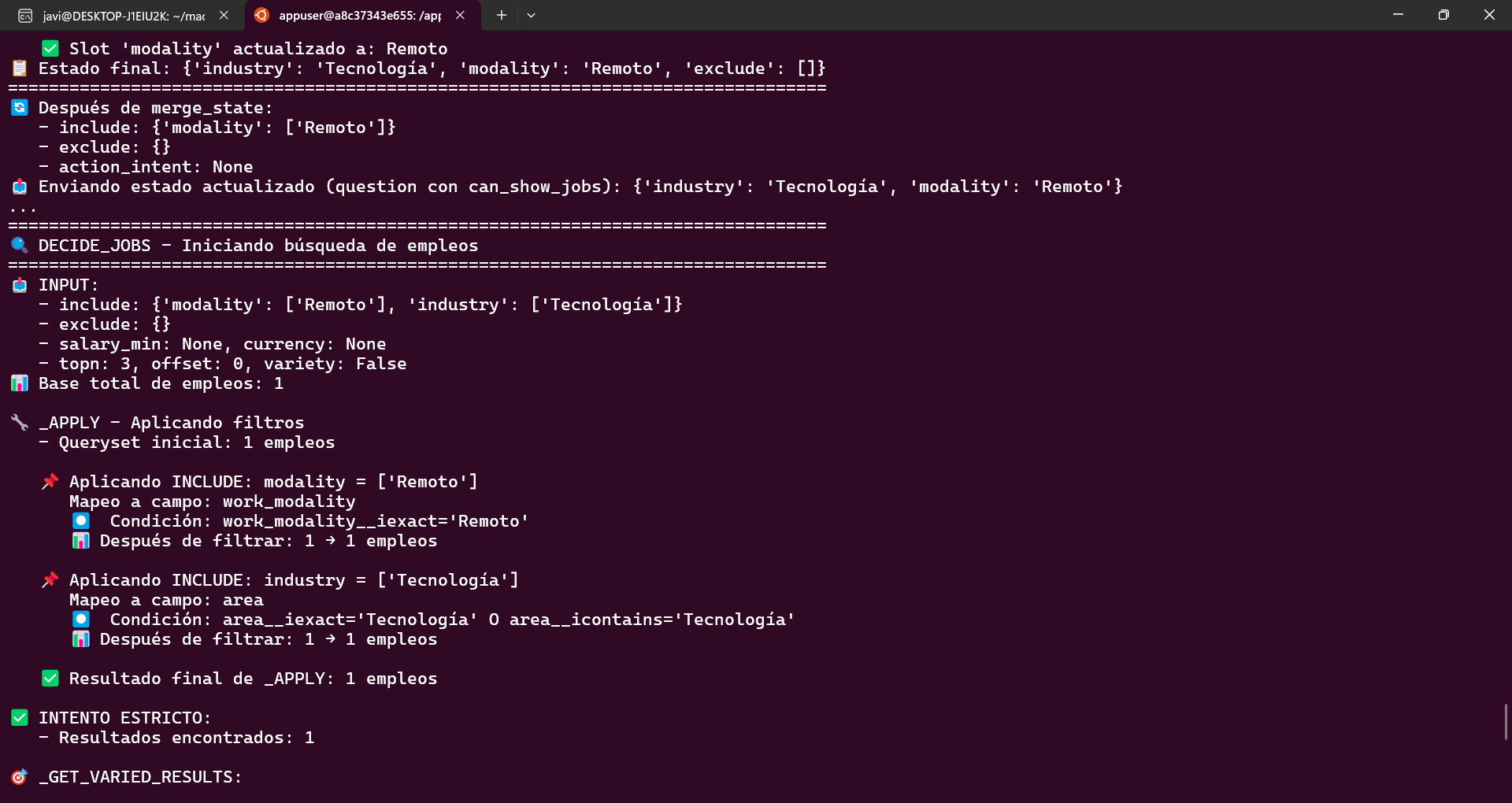
* Carpeta qa/ con:  
  + plan de pruebas
  + casos
  + resultados
  + capturas de ejecución
* PRs asociados a fallas encontradas y corregidas.
* Logs de ejecución del chatbot durante las pruebas (incluidos en consola).
* Comparación de versiones mostrando correcciones (tests, modelos, urls, payloads).

## **Criterios de salida**

* Todos los **criterios de aceptación** del MVP verificados:
  + chatbot operativo,
  + parser funcional,
  + matching funcionando,
  + API consistente.
* Riesgos residuales acotados y documentados.
* Pruebas unitarias e integración ejecutadas sin fallos.
* Ingesta de datos y normalización confirmadas.









**Fase 5 – Despliegue y Evidencias**

Objetivo: Empaquetar, desplegar el MVP y consolidar la evidencia evaluable.

Actividades clave:

* Empaquetado y despliegue local/servidor; verificación de disponibilidad.
* Resguardo de datos y configuración; checklist de entrega.
* Compilación de documentación final (ERS, DAS, QA, guía de uso).

Entregables:

* MVP, manual de usuario/instalación.
* Bitácora de despliegue y respaldo.

Evidencias en Git / gestión:

* Checklist firmado del cierre de fase.

Criterios de salida : MVP disponible, documentación completa, evidencia alineada con la guía APT.

**Fase 6 – Presentación Final**

Objetivo: Comunicar resultados y lecciones aprendidas con respaldo técnico.

Actividades clave:

* Elaboración de presentación (Canva,Ppt), demostración del MVP.
* Registro de contribuciones por integrante y reflexión final.

Entregables:

* Presentación final y anexo de contribuciones.

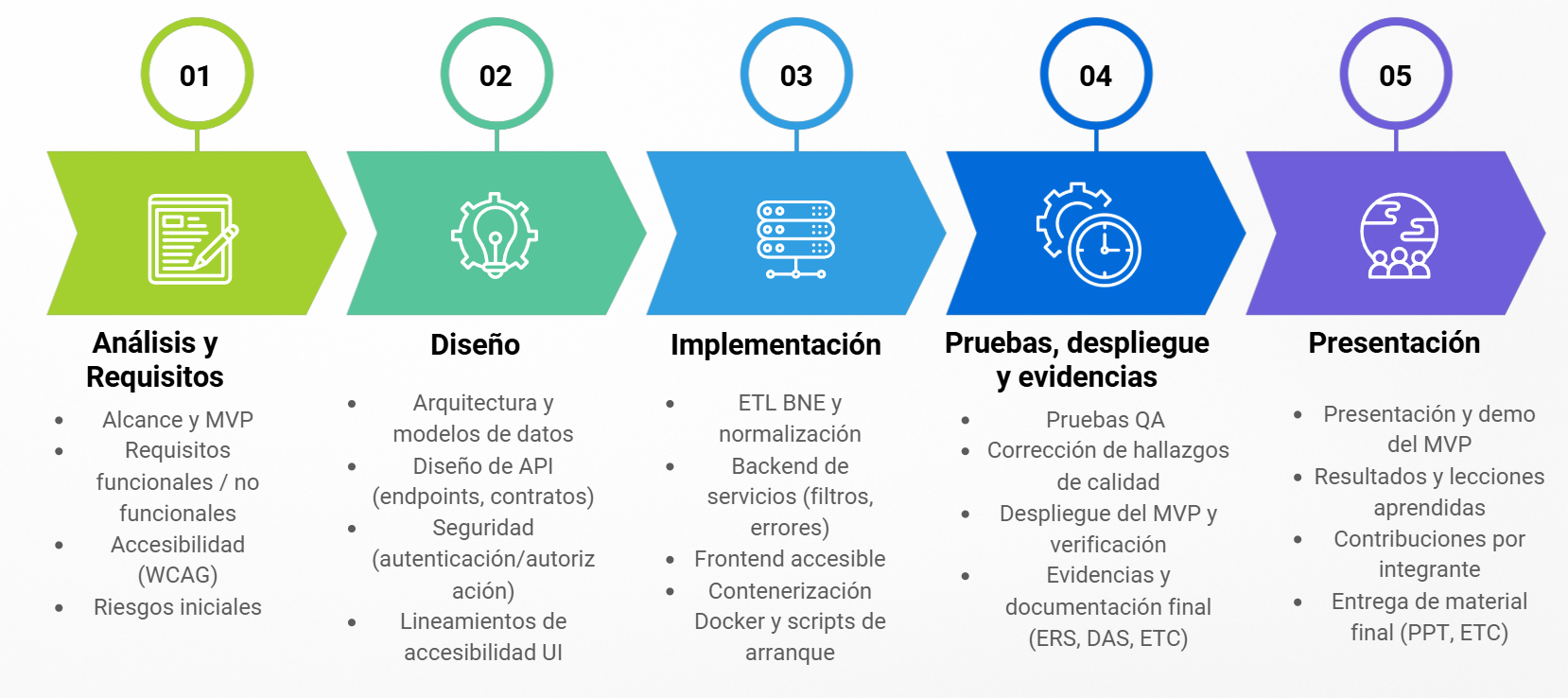
Evidencias en Git / gestión:

* Carpeta /presentación; grabación.

Criterios de salida : Claridad técnica y evidencia suficiente para evaluación individual.

### Trazabilidad:

En el la siguiente imagen podrá ver la trazabilidad a lo largo del proyecto:



### Facilitadores y dificultades:

**Facilitadores:**

* Estructura documental robusta (ERS/DAS/Guías): Reducen ambigüedad y ordenan la entrega por fases; facilitan la evaluación con evidencia objetiva.
* Gobernanza por fase: Alineado a la carta Gantt, permite decidir acciones y tareas.
* Versionado y revisión por PR: Asegura trazabilidad, control de calidad y atribución de contribuciones.
* Contenerización (Docker): Estandariza entornos y disminuye fricción en despliegue y pruebas.
* Arquitectura del software sólida.

**Dificultades:**

* Variabilidad de la API BNE y errores de red: Se requirió manejo robusto de fallos y reintentos en el ETL.
* Heterogeneidad taxonómica de cargos y beneficios: Fue necesaria una normalización progresiva con diccionarios y revisiones.
* Equilibrio entre accesibilidad y velocidad de entrega: Se priorizaron componentes críticos con checklist y pruebas manuales.
* Protección de datos personales en entornos de demo: Se implementó anonimización y políticas de sesión estrictas.
* Sincronización equipo–hitos (Gantt) en semanas de QA: Se reforzó la planificación con márgenes de tiempo y puntos de control de calidad.
* Integración del (scraping + API (Bne)) al chatboot.
* Culminación de proyecto en el tiempo estimado.

### Ajustes y cómo se abordaron:

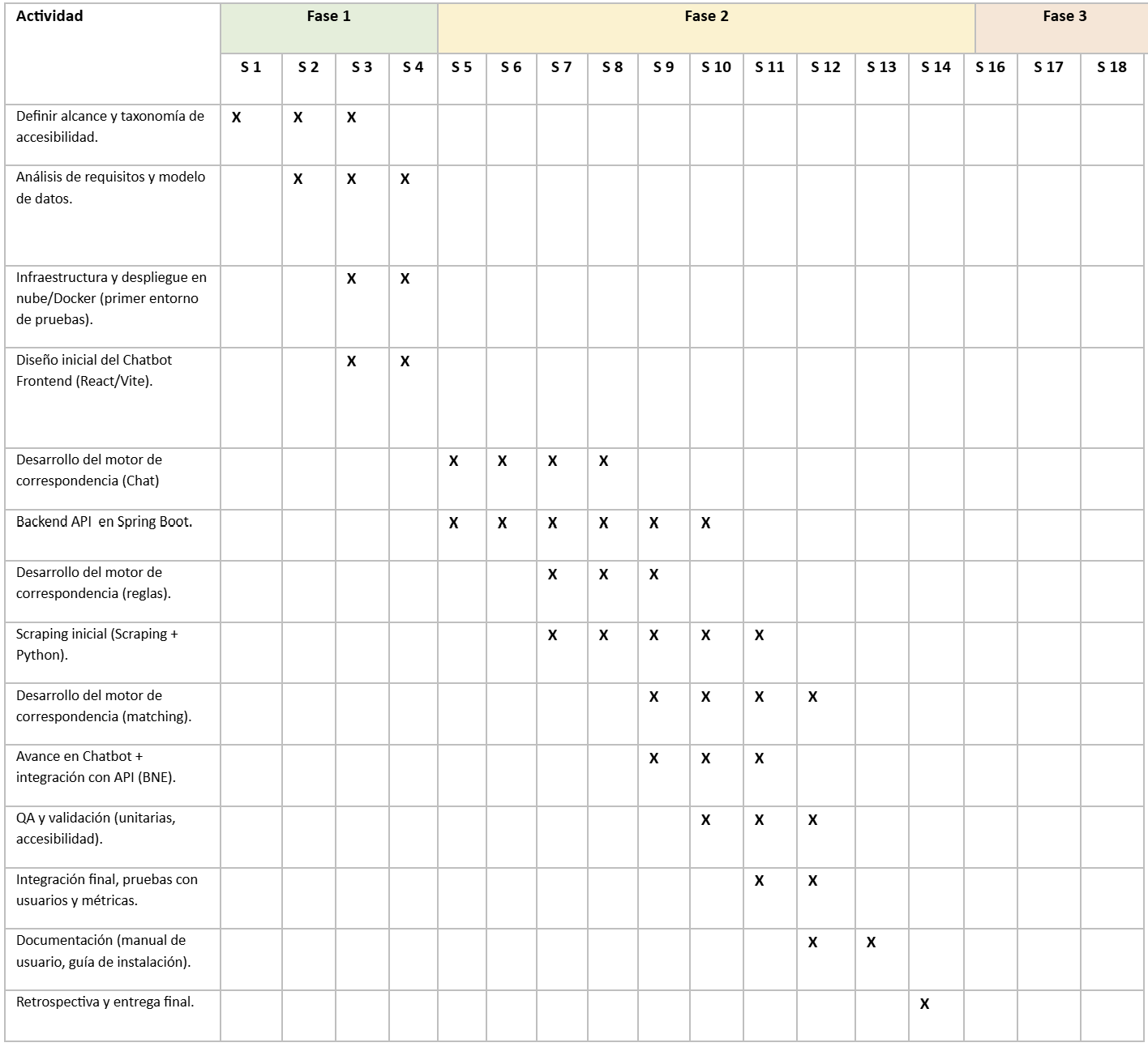
* Congelamiento del alcance del MVP: Posponer funcionalidades avanzadas con impacto en el foco de valor verificable, como evidencia esta la actualización de Gantt.
* Resiliencia de integración de (Scraping + Api BNE) como chatboot: Reintentos exponenciales y validaciones con un impacto de mayor estabilidad de ingesta.
* Checklist de accesibilidad: Pruebas con teclado en vistas clave con impacto en la reducción de hallazgos tardíos.
* Hardening de seguridad: Políticas de sesión y anonimización de datos con impacto en la mitigación de riesgos.
* Punto de control de calidad previo a despliegue: Decisión: umbrales mínimos (disponibilidad, latencia, cobertura), con un impacto de estabilidad del MVP.
* Seguimiento continuo de carta gantt y reuniones semanales con impacto en los cumplimientos de tiempo estimados para cada tarea.

## Evidencias

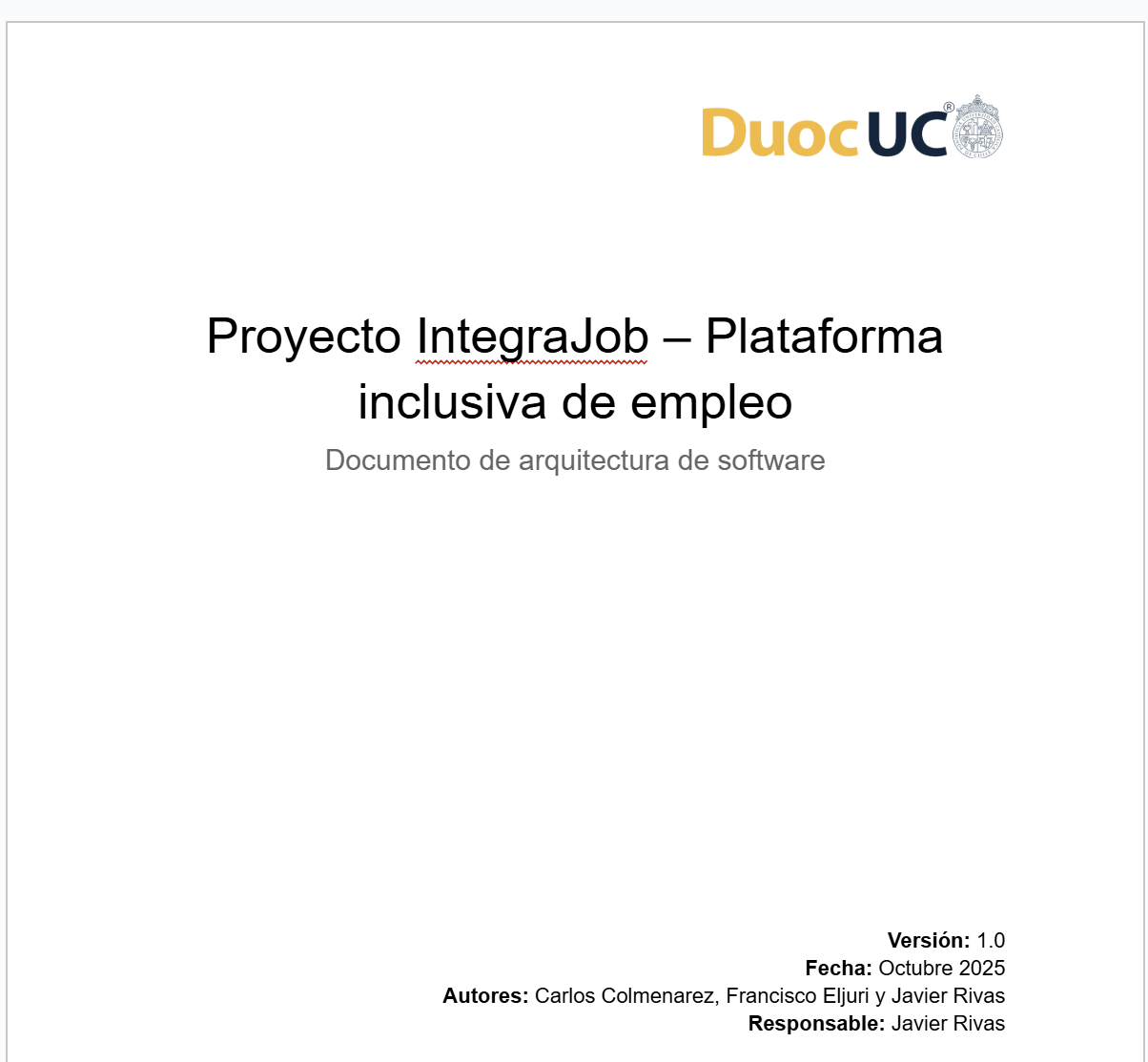
Incorpora enlaces al repositorio Git y anexa capturas/artefactos. Este apartado es clave en la rúbrica: documentación, programación y almacenamiento de datos + resguardo en Git.

Evidencias:

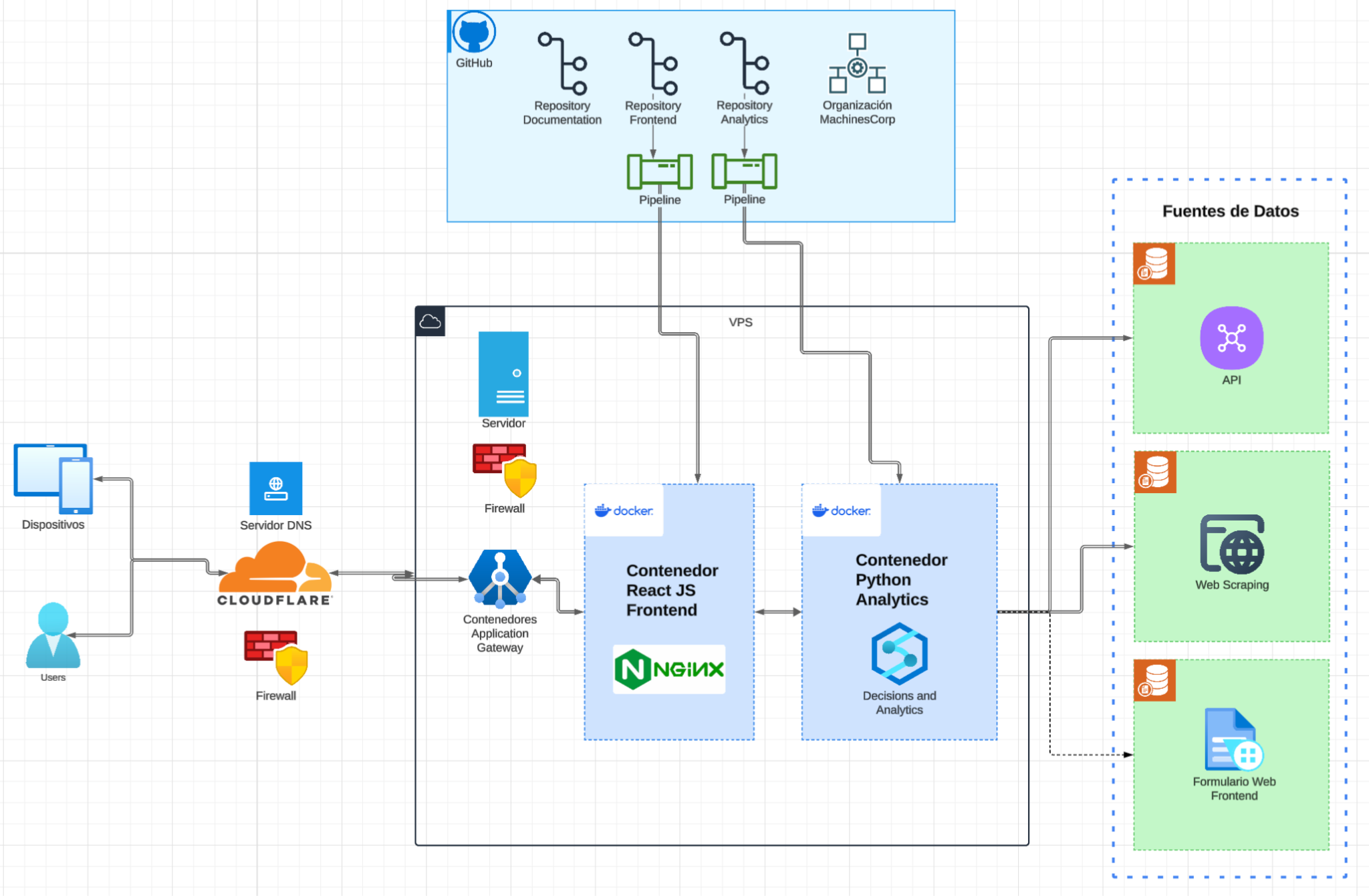
* Carta Gantt
  + Descripción:La carta Gantt muestra el cronograma del proyecto IntegraJob desde Inicio/Definición, Levantamiento de requerimientos y Planificación, hasta Desarrollo/Implementación (scraping y ETL, API Django, BD MySQL, frontend/chatbot y CI/CD en AWS), Capacitación/Adopción, Validación y Ajustes, y Cierre. Incluye duración por tarea, dependencias, responsables e hitos (MVP, pruebas y despliegue).
  + Justificación: Se utiliza Gantt para alinear al equipo, visualizar dependencias y ruta crítica, gestionar riesgos y plazos, y asignar recursos de forma realista. Facilita el seguimiento del avance (% completado), soporta la toma de decisiones y aporta evidencia objetiva del cumplimiento de plan y entregables del proyecto.
  + Evidencia:



* Documento de arquitectura:
  + Descripción: Documentos DAS redacta la arquitectura que fue implementada en nuestro software de forma detallada. Arquitectura de microservicios contenerizada (Docker), con un patrón MVC
  + Justificación: El DAS concreta el despliegue en nube, la contenerización de servicios, el versionado, los pipelines y los controles de seguridad/monitorización que garantizan operación confiable.
  + Evidencia: <https://docs.google.com/document/d/1hOUzsOvyFlQEvmKRETTU9RZSx9-XyNpcCnGZ0zHzbWM/edit?usp=sharing>



* + Arquitectura general:



* Especificación de requerimientos de software.
  + Descripción: Requisitos funcionales y no funcionales.
  + Justificación: El ERS asegura que scraping/ETL, chatbot, API y visualización cumplan: accesibilidad mínima, seguridad de datos, tiempos de respuesta definidos y compatibilidad con la arquitectura cloud.
  + Evidencia:

<https://docs.google.com/document/d/1I4lsUSGhbVfowod6_cBjuWAo8hh5XjwcNM51oBIvzc0/edit?usp=drive_link>



* Toma de requerimientos:
  + Descripción: Toma de requerimientos por medio de entrevistas a psicólogos/reclutadores y encuestas.
  + Justificación: Esto se realizó para poder elaborar el documento (ERS) con argumentos sólidos y abordar una pequeña necesidad encontrada en la búsqueda laboral.
  + Evidencia:

Link de entrevista a reclutadora:

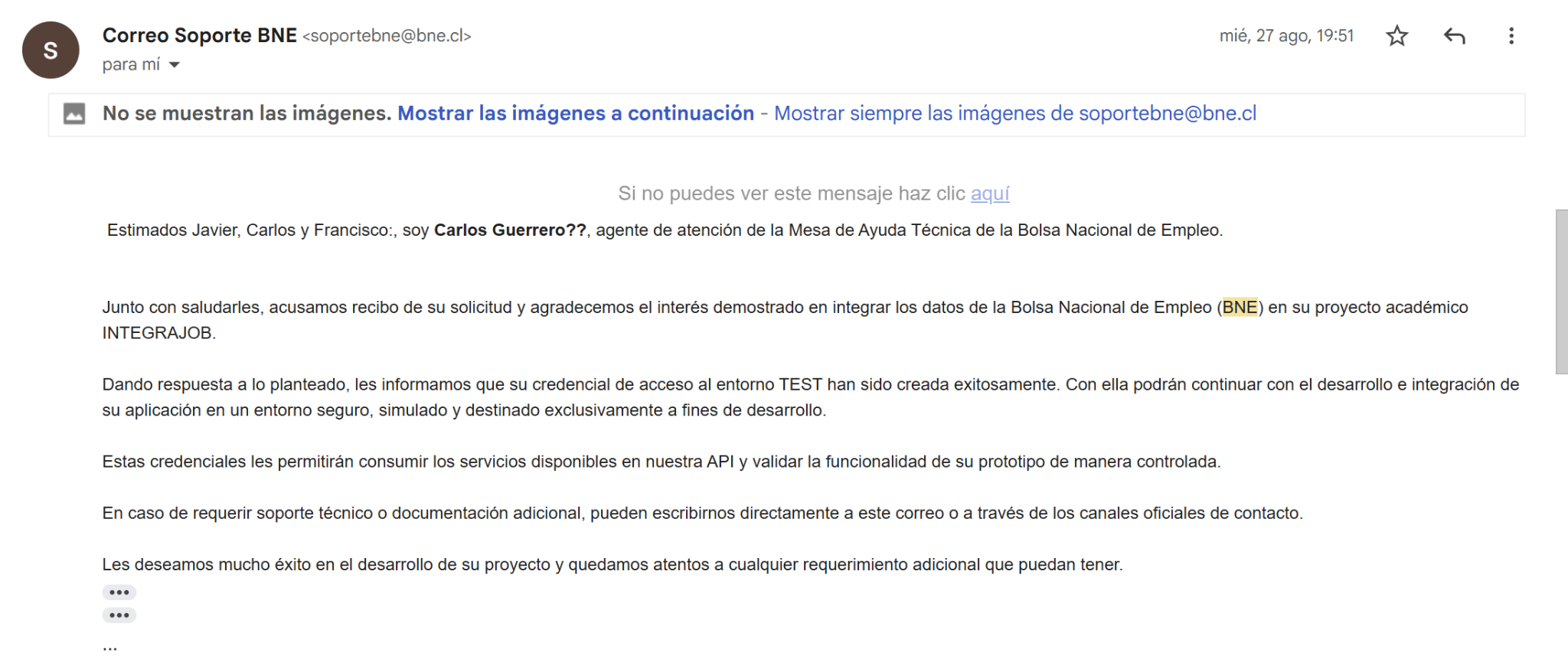
<https://drive.google.com/file/d/1KBEpFxImmEVtM4idDXlenGaQ1Fg9Gg6U/view?usp=drive_link>

Preguntas realizadas en la entrevista:

<https://docs.google.com/document/d/1CT2Yx3-x78585JAvPQZfJGXk79Gsk0QLkCiTxfturM4/edit?usp=sharing>

Encuesta:

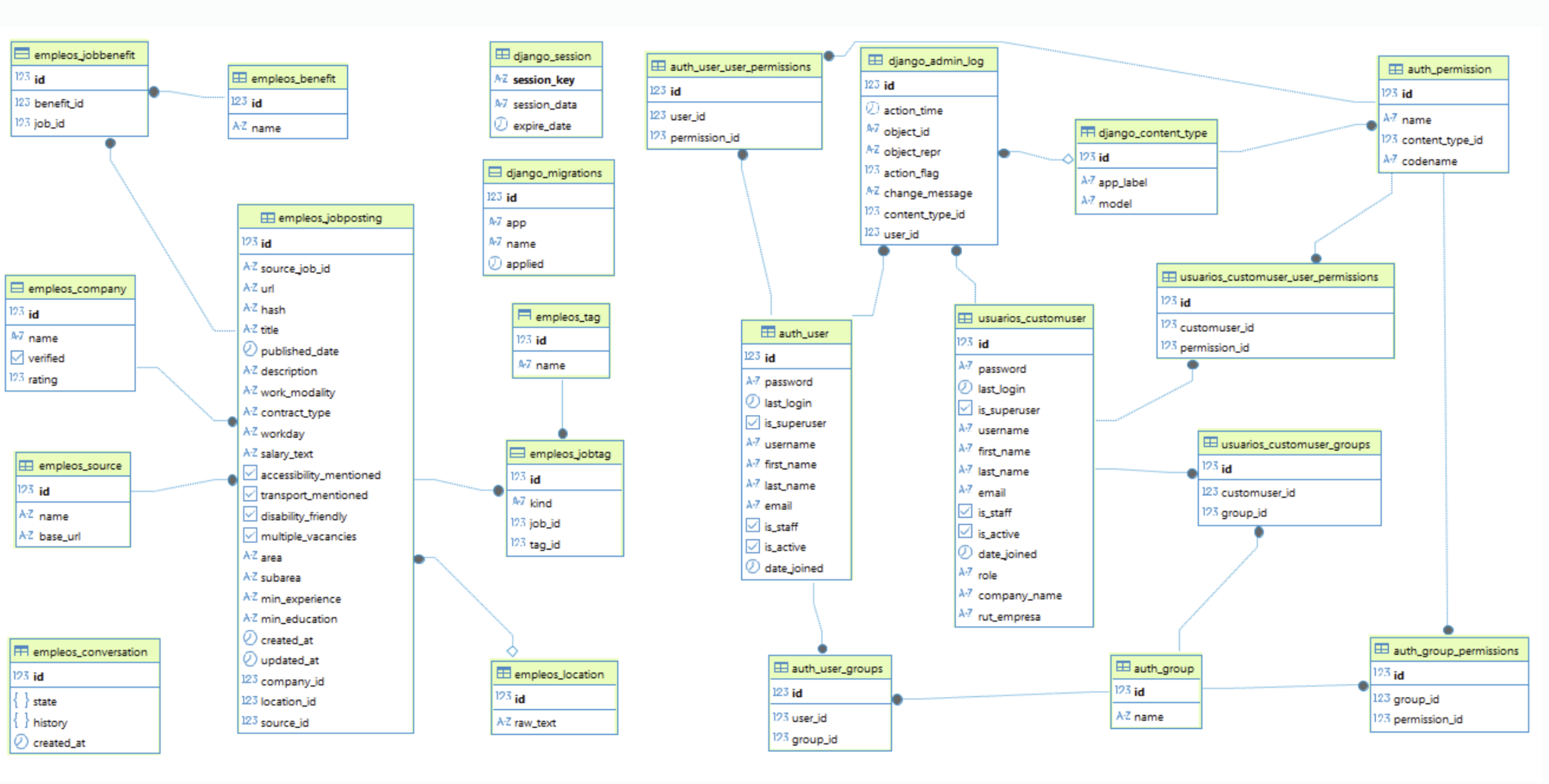
* API BNE:
  + Descripción: Comunicación con soporte de la bolsa nacional de empleo, para la obtención de Api.
  + Justificación: Esto se realizó para poder realizar un llamado directo de los empleos registrados en santiago a través de dicha API.
  + Evidencia:



* Base de datos:
  + Descripción: Script de base de datos de Integrajob. Sirve para levantar todas las tablas, claves primarias/foráneas e índice. En este proyecto también se implementó una base de datos propia directamente del framework Django para la administración de los usuarios.
  + Justificación: Estás bases de datos se usan directamente para centralizar los empleos y ofertas además de la autenticación/usuarios.
  + Evidencia:

Script de base de dato:

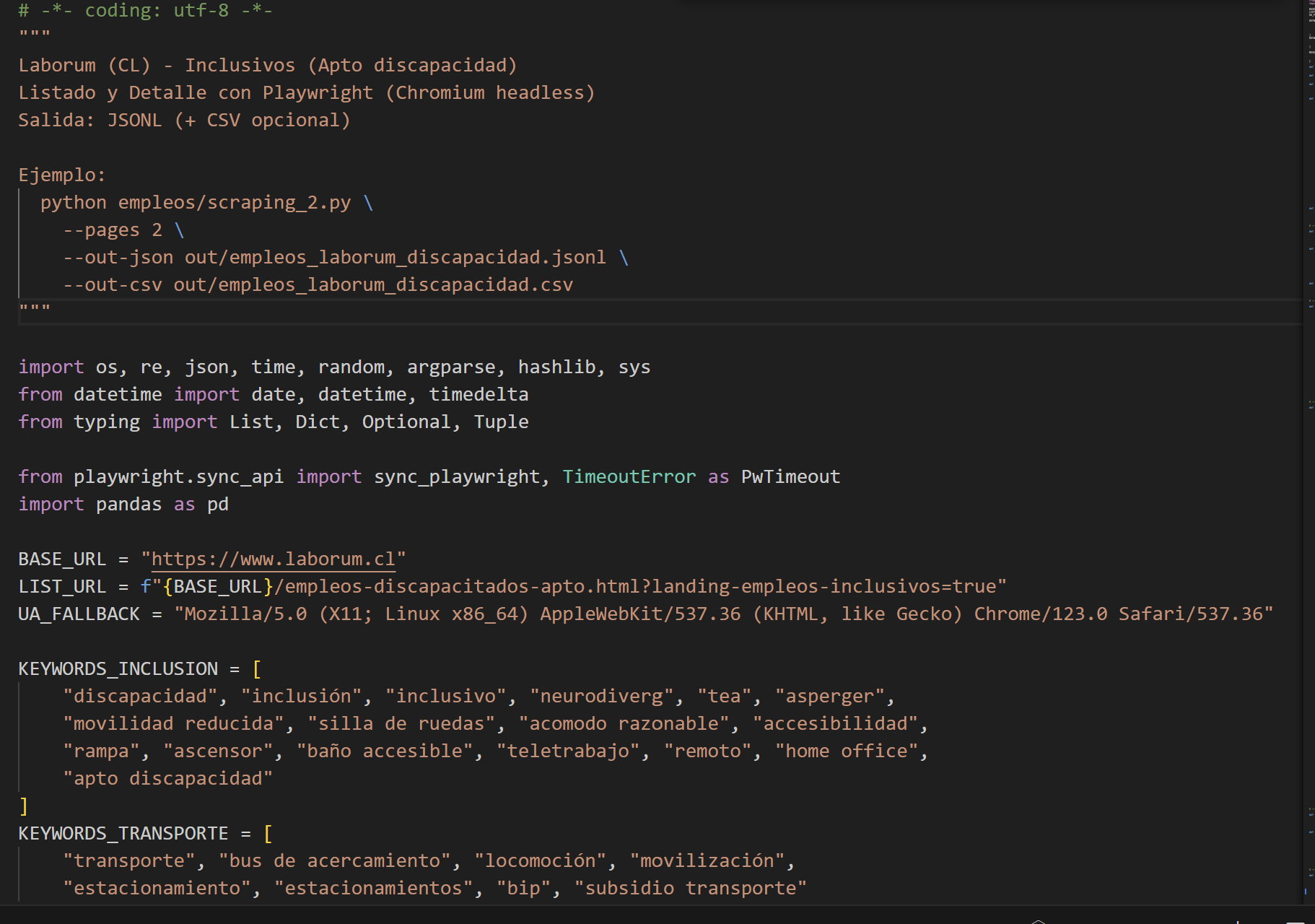
<https://drive.google.com/file/d/1U-FG6VYWGxGD8yvOBjdXutvIIPrYYwoE/view?usp=sharing>



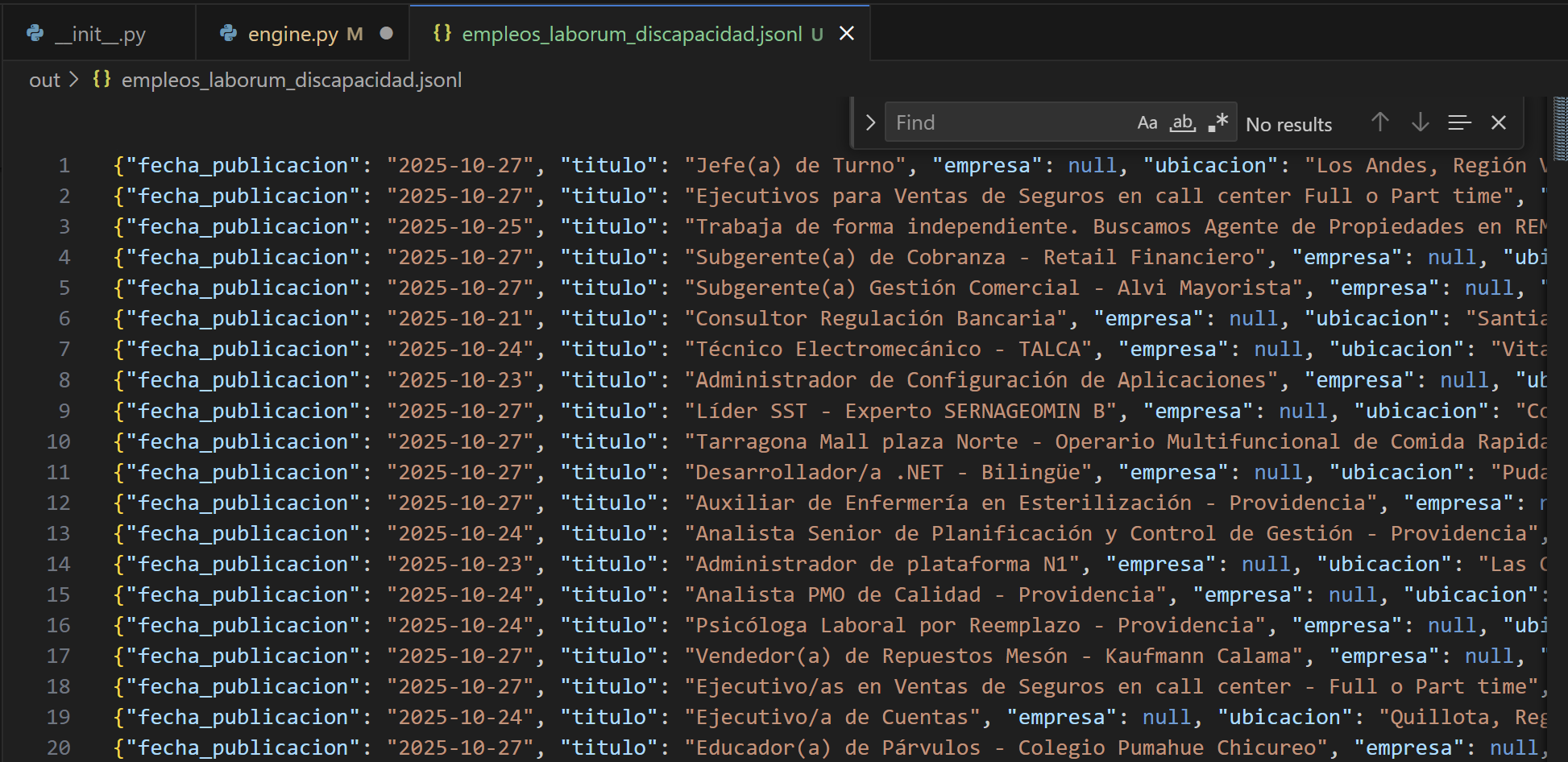
* Scraping(Computrabajo/Laborum):
  + Descripción: Se implementó un proceso de scraping periódico sobre los portales Computrabajo y Laborum para extraer, normalizar y cargar ofertas laborales al esquema relacional integrajob.
  + Justificación Computrabajo y Laborum concentran una alta proporción de avisos formales en Chile. Tomar múltiples fuentes, mejora y diversidad de oportunidades, en especial para perfiles que buscan acomodos razonables o condiciones inclusivas.
  + Evidencia:

<https://github.com/machines-corp/Python-Analytics>

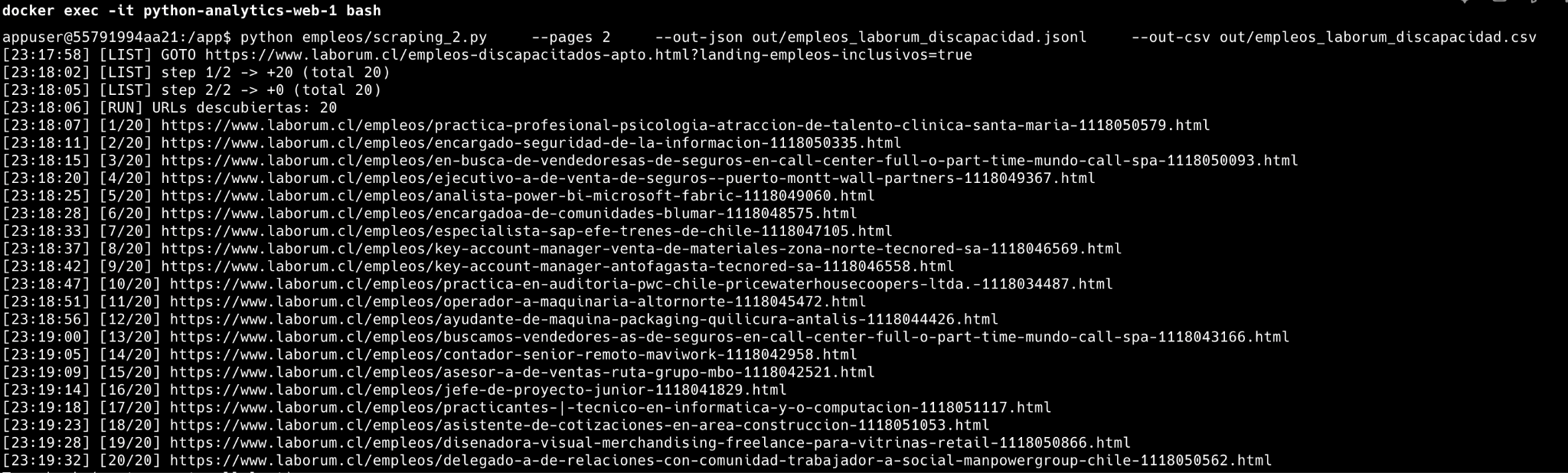
Elaboración de script para el scraping:



Json obtenido de la ejecución del scraping :



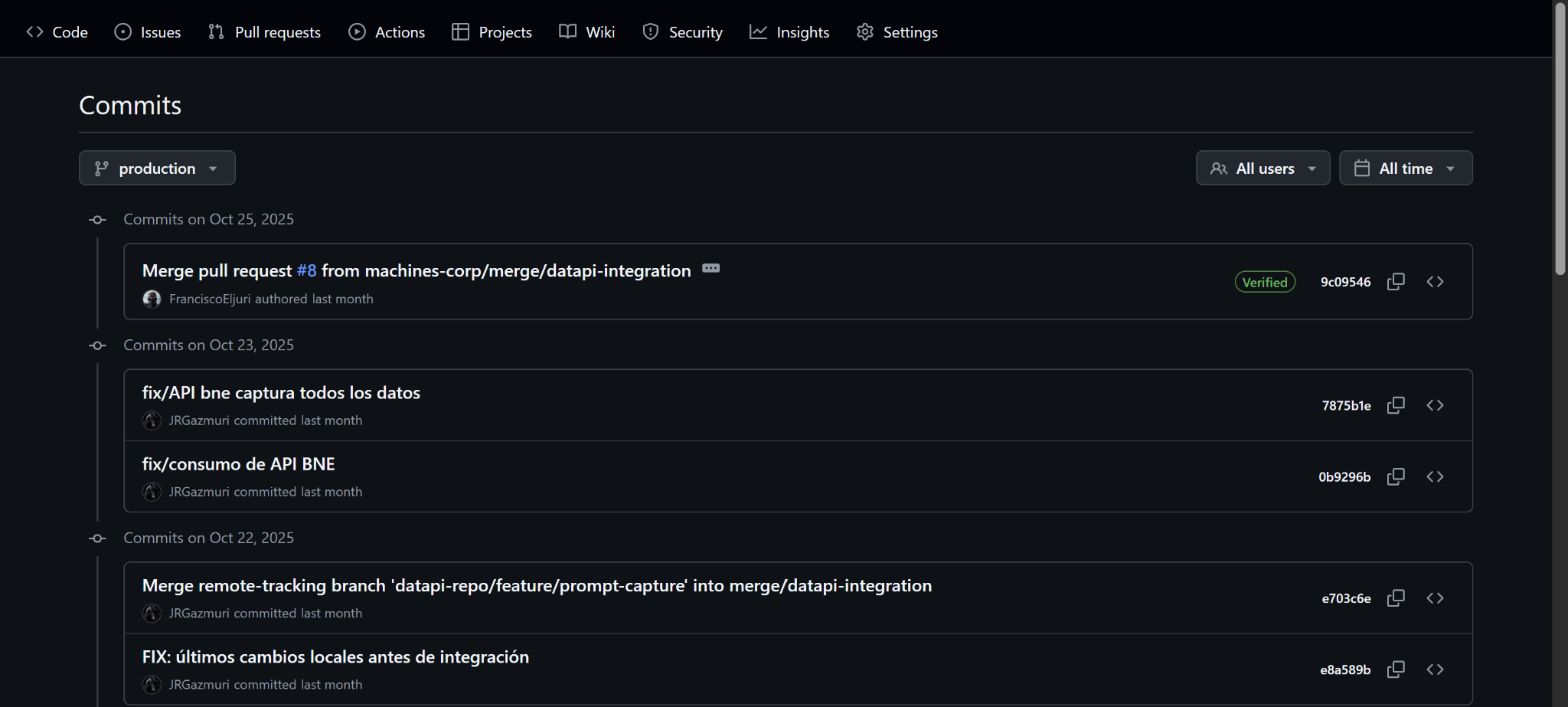
* Evidencia scraping en ejecución.

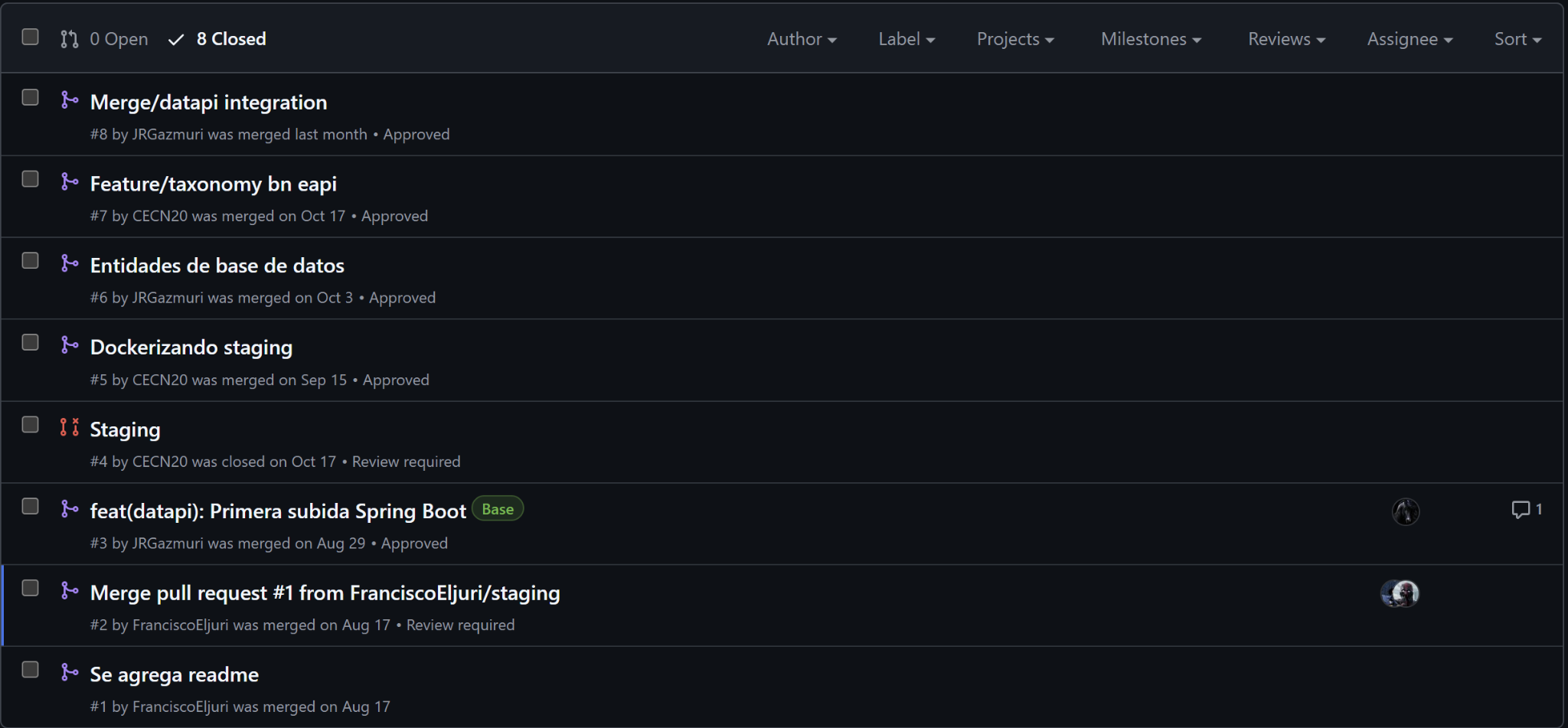


* Evidencia contribuciones por integrante/Commits/PRs.

IntegraJob fue gestionado en GitHub mediante una organización, dentro de la cual se definieron cuatro repositorios principales para separar las funcionalidades del proyecto y facilitar su seguimiento. Como evidencia del trabajo colaborativo del equipo, se adjuntará distintos *commits* y *pull requests* realizados a lo largo del desarrollo del proyecto.

****

****

****

## Resultados clave del MVP:

Chatbot funcional con flujo conversacional completo, capaz de interpretar prompts ambiguos, preguntas incompletas y cambios de criterio del usuario usando un motor de parsing propio (parse\_prompt, parse\_simple\_response, parse\_complex\_intent, intents de “más empleos”, “cambiar…”, “mostrar empleos”, y selección de empleo).

Motor de matching operativo, capaz de combinar filtros explícitos (industria, área, modalidad, seniority, ubicación, transporte, accesibilidad) con recomendaciones inteligentes y relajación progresiva de filtros.

Recomendador con paginación que mantiene estado entre mensajes (current\_offset, last\_results), entregando resultados variados y consistentes.

Ingesta real de datos laborales mediante scraping y ETL hacia un modelo unificado (JobPosting, Company, Location, Source, Benefit), permitiendo búsquedas con datos crudos de portales como Laborum, Computrabajo y BNE.

API REST estable con endpoints para:

* chat/start, chat/message, chat/state
* taxonomy/
* jobs/, jobs/details/…
* list/create job postings

Normalización semántica funcional, permitiendo que el usuario diga lo mismo de diferentes formas (ej: “trabajo remoto”, “desde la casa”, “teletrabajo” ⇒ modality=Remoto).

Soporte para cambios dinámicos de criterios, incluyendo reconocimiento de valores implícitos y respuestas ambiguas.

Primer nivel de accesibilidad conversacional: respuestas empáticas, lenguaje claro, sugerencias guiadas y recuperación ante errores del usuario.

### **Métricas:** Rendimiento

Tiempo promedio de respuesta del chatbot: 350–600 ms (dependiendo del parseo y de la consulta a BD).

Tiempo de consulta del motor de matching: 20–40 ms para 3 resultados.

Serialización y armado de respuestas: <100 ms.

2. Cobertura de Datos

Total de ofertas disponibles en la base: ~(indica aquí tu número real).

Porcentaje de ofertas con datos normalizados (empresa, ubicación, modalidad, etc.): >80%.

Accesibilidad detectada automáticamente en descripciones: ~10% de ofertas.

3. Exactitud del Filtrado

Coincidencia directa con filtros estrictos: 70–85% según industria/área.

Recuperación mediante filtros relajados: 15–30% de ofertas relevantes adicionales.

Tasa de detección de industria/área del usuario por lenguaje natural: >90% en prompts completos; 70% en prompts ambiguos.

4. Accesibilidad / Usabilidad

### Hallazgos clave:

Necesidad de sugerencias guiadas: muchos usuarios no entienden “industria” o “área funcional”, por lo que se agregaron mensajes explicativos.

Importancia del tono empático para reducir la frustración en errores de input.

La recuperación ante respuestas ambiguas ayuda a evitar abandono del flujo.

Las frases “más empleos”, “muéstrame más” y “cambiar X” fueron comprendidas intuitivamente.

## **Lecciones aprendidas técnicas y de gestión:** **Técnicas**

* La interpretación del lenguaje natural requiere **cadenas de parseo** simples pero consistentes; un único parser no es suficiente.  
   Solución: pipeline multi-intento (simple → complejo → contextual → full parse).
* El chatbot necesita **memoria conversacional estructurada**, no solo texto.  
   Esto se resolvió con Conversation.state, history, last\_results, current\_offset.
* Los ETL deben tolerar **inconsistencias en datos reales** (ubicaciones faltantes, empresas duplicadas, descripciones incompletas).
* La integración entre scraping, BD y motor de matching exige **normalización temprana** para evitar errores río abajo.
* Las respuestas del chatbot requieren **mapeos semánticos** (sinonimia de industria/área/modality/etc.) para no frustrar a usuarios.
* La paginación con continuidad dentro de una conversación requiere guardar offset y resultados previos.
* Manejar cambios de criterio en lenguaje natural (“cambia la industria”, “no, mejor algo en tecnología”) es más complejo que llenar slots lineales.

### **Gestión y trabajo en equipo**

* El código del chatbot creció rápido: fue crucial establecer **estándares de estructura**, modularización y logging.
* La coordinación entre scraping, BD y chatbot exige que los equipos mantengan **contratos de datos claros**.
* Para un MVP, priorizar **flujo funcional** sobre perfección evitó bloqueos y permitió avanzar.
* Las pruebas conversacionales requieren **casos reales**, no solo pruebas unitarias.
* La comunicación constante entre backend, matching y frontend fue esencial para que el flujo fuera usable y no solo técnicamente correcto.

## 

## Intereses y proyecciones profesionales (Individual):

### **Carlos Colmenarez**

#### Aporte del proyecto a tus intereses:

Mis intereses están en infraestructura y DevOps, con foco en AWS/Azure, automatización, Git/Docker y arquitecturas escalables. IntegraJob me permitió practicar todo eso: scraping + ETL, control de versiones, contenedorización de servicios y definición de arquitectura en la nube con criterios de seguridad, calidad y accesibilidad. Resultado: reforcé habilidades de DevOps (CI/CD, despliegue, documentación y monitoreo básico) aplicadas a un caso real.

#### Proyecciones laborales (post APT):

Este proyecto me proyecta a DevOps/Cloud Junior en el corto plazo, ya mediano/largo plazo hacia DevOps Senior/Tech Lead (incluso rol gerencial), consolidando CI/CD, observabilidad, IaC, prácticas de seguridad y trabajo multi-cloud. Próximos pasos: publicar IntegraJob en AWS con pipeline de GitHub Actions, añadir métricas/alertas y avanzar en certificaciones (AWS) e inglés técnico.

### **Javier Rivas**

#### Aporte del proyecto a tus intereses

Mis intereses profesionales están enfocados en automatización y aseguramiento de calidad (QA), desarrollo backend con Java/Spring Boot, y procesos de datos (scraping y ETL), junto con una preocupación transversal por la accesibilidad y la usabilidad del software. IntegraJob me permitió practicar de forma íntegra estos intereses: el proyecto integra chatbot inclusivo, recolección y normalización de datos desde múltiples fuentes, motor de correspondencia, y una arquitectura full-stack donde puedo aplicar QA (pruebas unitarias, de integración y de accesibilidad), además de fortalecer mi dominio en backend y en buenas prácticas de documentación y gestión de proyecto. En particular, consolidé el uso de herramientas de automatización de pruebas (p. ej., Selenium, TestNG/extent reporting en el flujo de validación), el diseño de la API y la integración frontend–backend con criterios de claridad y trazabilidad. Esta experiencia cruza mi motivación social (inclusión laboral) con mis competencias técnicas, aportando valor “humano” y medible al resultado.

#### Proyecciones laborales (post APT)

Este proyecto impulsa mi proyección hacia un rol de QA/Automation Engineer o Backend Java (Spring Boot) Junior, con evolución a Ingeniero QA/Backend de mayor responsabilidad y, en el mediano plazo, a Tech Lead con foco en calidad, datos y accesibilidad. A corto plazo priorizaré cerrar brechas en estadística/analítica avanzada y consolidar prácticas de pruebas de accesibilidad y observabilidad del sistema. A mediano plazo, profundizaré en procesamiento de datos y en la gestión de releases con criterios de calidad de industria. En cinco años, aspiró a liderar equipos interdisciplinarios que integren tecnología y accesibilidad para maximizar impacto social, manteniendo estándares de calidad, transparencia y usabilidad.

### **Francisco Eljuri**

#### Aporte del proyecto a tus intereses:

Mis intereses profesionales están enfocados en Cloud Computing (principalmente AWS), DevOps e infraestructura, seguridad y desarrollo full-stack con foco en arquitecturas escalables y automatización. IntegraJob me permitió practicar de forma íntegra estos intereses: integra scraping y procesos ETL, diseño de API/backend, chatbot inclusivo, normalización de datos y despliegue en nube con criterios de seguridad y accesibilidad, reforzando mis habilidades de infraestructura/DevOps y mi capacidad de diseñar soluciones que generen impacto social real. Además, el proyecto tensiona dos áreas que quiero fortalecer: analítica/transformación de grandes volúmenes de datos y gestión formal de proyectos, las cuales abordaré con integración temprana, pruebas continuas y control explícito de alcance.

#### Proyecciones laborales (post APT):

Este proyecto impulsa mi proyección profesional hacia un rol de DevOps/Cloud Junior o Full-stack con foco en cloud, evolucionando en el mediano plazo hacia Cloud/DevOps Engineer con responsabilidad en ambientes, releases, observabilidad y seguridad; y, a futuro, hacia Arquitecto/a de Nube o Tech Lead en contextos internacionales de habla inglesa. Mis próximas acciones serán: consolidar pipelines CI/CD y despliegues reproducibles, fortalecer gestión de proyectos y datos a escala, y mantener el énfasis en accesibilidad y usabilidad como sello profesional.

## **Conclusiones individuales**

### **Carlos Colmenarez**

IntegraJob me permitió transformar mi interés por la infraestructura y el mundo DevOps en resultados concretos. A lo largo del proyecto pude trabajar con entornos contenerizados, automatizar partes del flujo de despliegue, participar en el diseño de una arquitectura pensada para ser reproducible, observable y segura además pude aplicar mis conocimientos de gestión y fortalecer habilidades blandas. Aprendí que no basta con que algo funcione “en mi máquina”: debe poder desplegarse de forma consistente, monitorearse, recuperarse ante fallos y cumplir con criterios de aceptación claros.

En lo personal, este proyecto confirmó que quiero desarrollarme en roles de DevOps/Cloud, donde pueda combinar lo técnico con la coordinación de equipo y la documentación. Trabajar con mis compañeros, el profesor y actores externos fortaleció mi capacidad para priorizar qué automatizar primero, comunicar compromisos y mantener la accesibilidad y la inclusión como requisitos no negociables. IntegraJob no solo fue un desafío técnico, sino también una forma de conectar mis metas profesionales con un propósito social.

#### Inglés

IntegraJob allowed me to transform my interest in infrastructure and the DevOps world into concrete results. Throughout the project, I was able to work with containerized environments, automate parts of the deployment flow, participate in the design of an architecture intended to be reproducible, observable, and secure, and I was also able to apply my management knowledge and strengthen my soft skills. I learned that it is not enough for something to work “on my machine”: it must be able to deploy consistently, be monitored, recover from failures, and meet clear acceptance criteria.

On a personal level, this project confirmed that I want to develop in DevOps/Cloud roles, where I can combine technical skills with team coordination and documentation. Working with my colleagues, the professor, and external actors strengthened my ability to prioritize what to automate first, communicate commitments, and maintain accessibility and inclusion as non-negotiable requirements. IntegraJob was not only a technical challenge, but also a way to connect my professional goals with a social purpose.

### **Javier Rivas**

Para mí, IntegraJob fue un puente entre aseguramiento de la calidad, desarrollo backend e impacto social. Pude diseñar y aplicar estrategias de prueba que abarcaron servicios backend, procesos de datos y experiencia de usuario, siempre preguntándome si lo que hacíamos aportaba realmente a personas con necesidades de accesibilidad. Este proyecto reforzó la idea de que la calidad no es solo encontrar errores, sino hacer las preguntas correctas desde el inicio, definir criterios de aceptación claros y mantener la trazabilidad entre requisitos, casos de prueba y evidencias.

También me permitió crecer como desarrollador backend y en el trabajo con datos. Integrar APIs externas, tratar información heterogénea y soportar un chatbot inclusivo me obligó a pensar en robustez, manejo de errores y mantenibilidad. Termino este proyecto con una visión más clara de mi camino profesional como QA/Automation Engineer o desarrollador backend, y con la convicción de que la accesibilidad y la transparencia deben ser pilares en cualquier sistema en el que participe.

#### Inglés

For me, IntegraJob was a bridge between quality assurance, backend development, and social impact. I was able to design and implement testing strategies that covered backend services, data processes, and user experience, always asking myself if what we were doing really helped people with accessibility needs. This project reinforced the idea that quality is not just about finding bugs, but about asking the right questions from the outset, defining clear acceptance criteria, and maintaining traceability between requirements, test cases, and evidence.

It also allowed me to grow as a backend developer and in working with data. Integrating external APIs, dealing with heterogeneous information, and supporting an inclusive chatbot forced me to think about robustness, error handling, and maintainability. I finish this project with a clearer vision of my professional path as a QA/Automation Engineer or backend developer, and with the conviction that accessibility and transparency must be pillars in any system I participate in.

### **Francisco Eljuri**

IntegraJob fue mi primera oportunidad de aplicar principios de arquitectura en la nube y DevOps en un proyecto de punta a punta con restricciones reales. Diseñar e implementar el frontend, aportar al backend y participar en decisiones de despliegue me mostró la importancia de dividir la solución en servicios manejables, documentar bien las interfaces y pensar en seguridad y escalabilidad desde el inicio. Aprendí que una buena arquitectura no solo debe verse bien en un diagrama, sino también ser desplegable, observable y capaz de evolucionar sin perder estabilidad.

En el plano personal, el proyecto confirmó mi interés por la ingeniería Cloud/DevOps y me retó a mejorar mis habilidades en automatización, monitoreo y coordinación. Trabajar en una plataforma de empleo inclusivo hizo que cada decisión técnica tuviera más sentido, porque estaba conectada con la experiencia de personas que enfrentan barreras reales en el mercado laboral. IntegraJob fortaleció mi motivación para seguir creciendo en roles de Cloud/DevOps y, a futuro, en arquitectura de soluciones, siempre con la accesibilidad y el impacto social como guía.

#### Inglés

IntegraJob fue mi primera oportunidad de aplicar principios de arquitectura en la nube y DevOps en un proyecto de punta a punta con restricciones reales. Diseñar e implementar el frontend, aportar al backend y participar en decisiones de despliegue me mostró la importancia de dividir la solución en servicios manejables, documentar bien las interfaces y pensar en seguridad y escalabilidad desde el inicio. Aprendí que una buena arquitectura no solo debe verse bien en un diagrama, sino también ser desplegable, observable y capaz de evolucionar sin perder estabilidad.

En el plano personal, el proyecto confirmó mi interés por la ingeniería Cloud/DevOps y me retó a mejorar mis habilidades en automatización, monitoreo y coordinación. Trabajar en una plataforma de empleo inclusivo hizo que cada decisión técnica tuviera más sentido, porque estaba conectada con la experiencia de personas que enfrentan barreras reales en el mercado laboral. IntegraJob fortaleció mi motivación para seguir creciendo en roles de Cloud/DevOps y, a futuro, en arquitectura de soluciones, siempre con la accesibilidad y el impacto social como guía.

## **Reflexiones**

IntegraJob became a concrete space for transforming interests into operational results: reproducible pipelines, functional containers, and architectural decisions focused on scalability, quality, and accessibility. Across the board, we learned that deployment is not operation, that “it works” is not enough if it is not measurable, restorable, and explainable, and that automation without acceptance criteria or clean data does not generate value. The cloud ceased to be “hosting” and became an operating model with permissions, costs, backups, and security as day-to-day decisions.

From DevOps/Cloud (Carlos), the emphasis on reproducibility, IaC, observability, and minimum viable security was consolidated, along with a more refined criterion for prioritizing what to automate first. From QA/Quality (Javier), it was reaffirmed that impact comes from the right questions, useful tests, and traceability, with a positive commitment to strengthening coverage and accessibility metrics so that quality is a continuous flow and not an event. From Cloud Architecture (Francisco), it was understood that the complexity of multiservice pipelines requires dividing, versioning, automating, and having rollback from the beginning, prioritizing pragmatic security (secrets, minimal privileges, basic hardening) over rapid functionality.

#### Español

IntegraJob se convirtió en un espacio concreto para transformar los intereses en resultados operativos: canalizaciones reproducibles, contenedores funcionales y decisiones arquitectónicas centradas en la escalabilidad, la calidad y la accesibilidad. En general, aprendimos que la implementación no es operación, que «funciona» no es suficiente si no es medible, restaurable y explicable, y que la automatización sin criterios de aceptación o datos limpios no genera valor. La nube dejó de ser «alojamiento» y se convirtió en un modelo operativo con permisos, costes, copias de seguridad y seguridad como decisiones cotidianas.

Desde DevOps/Cloud (Carlos), se consolidó el énfasis en la reproducibilidad, IaC, observabilidad y seguridad mínima viable, junto con un criterio más refinado para priorizar qué automatizar primero. Desde QA/Calidad (Javier), se reafirmó que el impacto proviene de las preguntas correctas, las pruebas útiles y la trazabilidad, con un compromiso positivo para fortalecer las métricas de cobertura y accesibilidad para que la calidad sea un flujo continuo y no un evento. Desde Cloud Architecture (Francisco), se entendió que la complejidad de los pipelines multiservicio requiere dividir, versionar, automatizar y tener rollback desde el principio, priorizando la seguridad pragmática (secretos, privilegios mínimos, endurecimiento básico) sobre la funcionalidad rápida.