Acta de constitución de proyecto



Proyecto: *“ComparaYa”*

**Integrantes:**

Gustavo Cartes **Docente:**

Italo Navarrete Felix Cifuentes CID

Daniel Stari

**Contenido**

[Información del proyecto 3](#_heading=h.gjdgxs)

[Propósito y justificación del proyecto 3](#_heading=h.1fob9te)

[Descripción del proyecto entregables 4](#_heading=h.3znysh7)

[Entregables del Proyecto 5](#_heading=h.6gblwhi26aa4)

[Requerimientos de alto nivel 6](#_heading=h.2et92p0)

[Tecnologías utilizadas 8](#_heading=h.1t3h5sf)

[Metodología de desarrollo 9](#_heading=h.4d34og8)

[Arquitectura del software 10](#_heading=h.2s8eyo1)

[Objetivos 11](#_heading=h.17dp8vu)

[Costos del proyecto 12](#_heading=h.rt5vlv9d1jj7)

[Premisas y restricciones 13](#_heading=h.3rdcrjn)

[Riesgos iniciales de alto nivel 14](#_heading=h.26in1rg)

[Cronograma de hitos principales revisar 15](#_heading=h.lnxbz9)

[Lista de Interesados (Stakeholders) 16](#_heading=h.1ksv4uv)

[Requisitos de aprobación del proyecto 17](#_heading=h.dbtbslb5pjyy)

[Aprobaciones 18](#_heading=h.2jxsxqh)

# **Información del proyecto**

* 1. **Datos**

| Empresa / Organización | RueditasVeloces |
| --- | --- |
| Nombre Proyecto | ComparaYa |
| Fecha de preparación | 11/08/2025 |
| Cliente | Duoc Uc |
| Patrocinador principal | Felix Cifuentes Cid |

# **Propósito y justificación del proyecto**

El propósito del proyecto es desarrollar una plataforma web que integre información oficial de precios de la Oficina de Estudios y Políticas Agrarias (ODEPA) con datos actuales obtenidos mediante técnicas de web scraping en portales de supermercados. La plataforma, construida con backend en Django y base de datos SQLite, permitirá procesar y visualizar la evolución de los precios mayoristas y minoristas a través de dashboards y gráficos interactivos, además de ofrecer una consulta centralizada de precios actuales de la canasta básica según cada supermercado.

Con ello, se busca entregar a los usuarios una herramienta tecnológica accesible, confiable y dinámica que facilite la comparación, análisis y comprensión de la información de precios, promoviendo la transparencia y mejorando la toma de decisiones tanto a nivel individual como institucional.

# **Descripción del proyecto entregables**

El proyecto consiste en el diseño y desarrollo de una plataforma web que integre datos oficiales y datos obtenidos en tiempo real para ofrecer una visión completa sobre los precios de alimentos en Chile. El sistema tendrá como núcleo un **backend en Django** y una **base de datos Postgresql**, encargados de procesar y almacenar los registros históricos provenientes de ODEPA.

La plataforma dispondrá de **dashboards interactivos** que mostrarán la evolución de precios mayoristas y minoristas mediante gráficos dinámicos, filtros y comparaciones personalizadas. Además, se incorporará un **módulo complementario de scraping** que permitirá recolectar información actualizada directamente desde sitios web de supermercados, con el fin de entregar al usuario una consulta centralizada y la posibilidad de comparar la canasta básica entre distintas cadenas.

El enfoque del proyecto contempla la **usabilidad** y **accesibilidad**, de manera que tanto consumidores individuales como pequeñas empresas, organizaciones públicas o privadas puedan utilizar la herramienta para fines de análisis, planificación y toma de decisiones estratégicas.

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

### 

# **Entregables del Proyecto**

1. **Documentación del diseño del sistema**

* Diagramas de arquitectura.
* Modelo entidad–relación de la base de datos.
* Descripción de módulos y componentes del backend.

1. **Backend funcional en Django con base de datos SQLite**

* Procesamiento y almacenamiento de datos históricos de ODEPA.
* API interna para la conexión con el frontend y los dashboards.

1. **Dashboards interactivos de precios**

* Visualización de la evolución de precios mayoristas y minoristas.
* Filtros por producto, fecha, región y tipo de mercado.
* Gráficos dinámicos (líneas, barras comparativas, indicadores).

1. **Módulo de scraping de supermercados**

* Extracción automática de precios actuales desde portales en línea.
* Actualización periódica de la base de datos con dichos valores.
* Visualización comparativa de la canasta básica entre supermercados.

1. **Interfaz web accesible**

* Panel de usuario con navegación simple e intuitiva.
* Diseño adaptable (responsive) para computadores y dispositivos móviles.

1. **Informe final del proyecto**

* Documentación técnica y funcional.
* Manual de usuario para consumidores y empresas.
* Conclusiones y recomendaciones para mejoras futuras.

# **Requerimientos de alto nivel**

* 1. **Requerimientos funcionales del producto**

| **ID** | **Requerimiento** |
| --- | --- |
| RENF001 | La plataforma debe contar con un backend desarrollado en Django. |
| RENF002 | El sistema debe importar y procesar datos oficiales de ODEPA (precios históricos). |
| RENF003 | La plataforma debe mostrar la evolución de precios mayoristas y minoristas mediante dashboards y gráficos interactivos. |
| RENF004 | Se debe implementar un módulo de scraping para obtener precios actuales desde sitios web de supermercados. |
| RENF005 | El sistema debe permitir comparar la canasta básica entre distintos supermercados. |
| RENF006 | La interfaz web debe ser accesible, amigable y adaptable a distintos dispositivos (responsive). |
| RENF007 | La plataforma debe permitir filtrar información por producto, fecha, tipo de mercado y supermercado. |
| RENF008 | El sistema debe actualizar periódicamente los datos recolectados del scraping. |
| RENF009 | El proyecto debe incluir documentación técnica y un manual de usuario. |

* 1. **Requerimientos no funcionales del producto**

| ID | Requerimiento |
| --- | --- |
| REF001 | Usabilidad: La interfaz debe ser intuitiva, simple y accesible para usuarios sin conocimientos técnicos. |
| REF002 | Compatibilidad: La plataforma debe ser responsive, adaptándose a computadores, tablets y smartphones. |
| REF003 | Mantenibilidad: El código debe estar documentado y estructurado siguiendo buenas prácticas de Django y Python. |
| REF004 | Rendimiento: El sistema debe mostrar dashboards y resultados de consultas en menos de 3 segundos en condiciones normales de carga. |
| REF005 | Seguridad: La plataforma debe proteger los datos contra accesos no autorizados mediante autenticación de usuarios. |
| REF006 | Escalabilidad: Aunque se inicie con SQLite, la arquitectura debe permitir migrar a bases de datos más robustas (PostgreSQL, MySQL) sin afectar la lógica del sistema. |
| REF007 | Confiabilidad: El sistema debe actualizar los datos de ODEPA y de supermercados sin pérdida ni duplicidad de información. |
| REF008 | Disponibilidad: La plataforma debe estar operativa al menos el 95% del tiempo en condiciones normales. |
| RENF009 | Portabilidad: El proyecto debe poder desplegarse en distintos entornos (local, servidor institucional o nube) con configuraciones mínimas. |
| RENF010 | Privacidad: Los datos de usuarios (si se implementa registro/login) deben ser gestionados conforme a buenas prácticas de protección de datos. |

# **Tecnologías utilizadas**

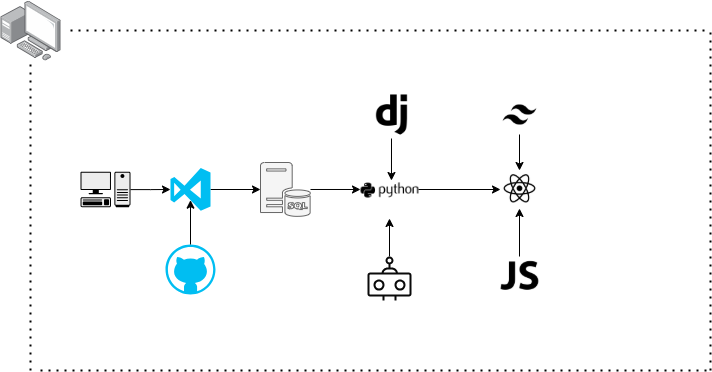
| **Categoría** | **Tecnología / Herramienta** | **Descripción** |
| --- | --- | --- |
| Lenguaje de programación | Python 3.x | Lenguaje principal para el desarrollo del backend y la lógica de negocio. |
| Framework Backend | Django | Framework web en Python que gestiona la lógica, APIs y conexión con la base de datos. |
| Base de Datos | SQLite | Base de datos ligera para almacenar los datos históricos de ODEPA y los recolectados por scraping. |
| Scraping / Recolección de datos | BeautifulSoup / Requests (Python) | Librerías para obtener precios actualizados desde sitios web de supermercados. |
| Framework Frontend | Vite + React | Vite permite un entorno de desarrollo rápido y optimizado; React gestiona la creación de componentes interactivos para el frontend. |
| Diseño y Estilo | TailwindCSS | Framework CSS para crear una interfaz moderna, responsiva y personalizable. |
| Lenguaje Frontend | JavaScript (ES6+) | Lenguaje base para la lógica del frontend, interacción con React y manejo de datos desde el backend. |
| Dashboards y Gráficos | Recharts / Chart.js | Librerías de visualización integradas en el frontend para mostrar la evolución y comparación de precios. |
| Control de versiones | Git + GitHub/GitLab | Gestión del código fuente y trabajo colaborativo. |
| Entorno de desarrollo | Visual Studio Code | Editor de código utilizado para la implementación del proyecto. |
| Gestión de dependencias | pip (Python) / npm (Node.js) | Herramientas para instalar y mantener librerías en backend y frontend. |

# **Metodología de desarrollo**

Para el desarrollo del proyecto se utilizará la metodología tradicional en cascada, ya que permite estructurar el trabajo de manera secuencial y ordenada, asegurando el cumplimiento de cada etapa antes de avanzar a la siguiente. Este enfoque es adecuado considerando que el proyecto se realizará de forma individual, lo que facilita la planificación lineal y la documentación clara de cada fase.

| **Actividad** | **Fase 1** | | | | **Fase 2** | | | | | | | | | | | | **Fase 3** | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **S 1** | **S 2** | **S 3** | **S 4** | **S 5** | **S 6** | **S 7** | **S 8** | **S 9** | **S 10** | **S 11** | **S 12** | **S 13** | **S 14** | **S 15** | **S 16** | | **S 17** | **S 18** |
| *Inicio de Capston* | **x** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |
| *Análisis del proyecto* |  | **x** |  |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |
| *Definición del proyecto y Diseño* |  |  | **x** |  |  |  |  |  |  |  |  |  |  |  |  |  | |  |  |
| *Análisis de datos* |  |  |  | **x** | **x** |  |  |  |  |  |  |  |  |  |  |  | |  |  |
| *Desarrollo Backend* |  |  |  |  |  | **x** | **x** | **x** |  |  |  |  |  |  |  |  | |  |  |
| *Desarrollo Frontend* |  |  |  |  |  |  |  |  | **x** | **x** | **x** |  |  |  |  |  | |  |  |
| *Integración de datos* |  |  |  |  |  |  |  |  |  |  |  | **x** | **x** | **x** |  |  | |  |  |
| *Gestión de proyecto* |  |  |  |  |  |  |  |  |  |  |  |  |  |  | **x** |  | |  |  |
| *Presentación* |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | **x** | | **x** | **x** |

# **Arquitectura del software**



# **Objetivos**

| **Objetivo** | **Indicador de Éxito** |
| --- | --- |
| Desarrollar una plataforma web que integre datos de ODEPA y supermercados. | Plataforma funcional accesible vía navegador, con conexión a BD y frontend operativo. |
| Implementar un backend en Django que procese y exponga datos mediante API. | API REST documentada y accesible que entregue datos de precios en formato JSON. |
| Construir dashboards interactivos que muestren la evolución de precios. | Dashboards disponibles con gráficos dinámicos y filtros aplicables por producto/fecha. |
| Desarrollar un módulo de scraping para obtener precios actuales de supermercados. | Ejecución periódica de scraping con datos actualizados en la BD sin duplicidad. |
| Permitir la comparación de la canasta básica entre supermercados. | Funcionalidad que muestre precios consolidados y comparativos en tiempo real. |
| Diseñar una interfaz responsiva e intuitiva con React, Vite y TailwindCSS. | Interfaz usable desde PC y dispositivos móviles, validada en pruebas de usuario. |
| Documentar el sistema y generar manual de usuario. | Entrega de documentación técnica y manual funcional al finalizar el proyecto. |

# **Costos del proyecto**

El proyecto contempla la participación de cuatro roles principales: Jefe de Proyecto (JP), Programador Back-End (PBE), Programador Front-End (PFE) y Administrador de Base de Datos y Servidores (DBA). Cada rol cuenta con un responsable definido y una tarifa horaria específica, lo que permite distribuir las tareas de manera clara y realista a lo largo de las fases del proyecto.

El cronograma de costos se ha estructurado en cuatro fases: Análisis y Planificación, Diseño, Desarrollo, e Implementación y Cierre. Cada fase considera un número estimado de horas por rol, lo que genera un costo parcial y, en conjunto, un costo total proyectado de $11.290.000 CLP.

Este desglose entrega visibilidad sobre el esfuerzo requerido por cada integrante, asegura trazabilidad en el uso de recursos y permite gestionar de forma eficiente el presupuesto asignado al proyecto.

| **Sigla** | **Rol** | **Nombre Responsable** | **Tarifa por Hora (CLP)** |
| --- | --- | --- | --- |
| JP | Jefe de Proyecto | Italo Navarrete | $40.000 |
| PBE | Programador Back-End | Gustavo Cartes | $35.000 |
| PFE | Programador Front-End | Daniel Stari | $35.000 |
| DBA | Administrador BD y Servidores | Italo Navarrete | $38.000 |

| **Fase** | **JP (hrs)** | **PBE (hrs)** | **PFE (hrs)** | **DBA (hrs)** | **Costo Total (CLP)** |
| --- | --- | --- | --- | --- | --- |
| Análisis y Planificación | 20h | 10h | 10h | 5h | $1.580.000 |
| Diseño | 25h | 20h | 20h | 10h | $2.430.000 |
| Desarrollo | 30h | 60h | 60h | 20h | $5.300.000 |
| Implementación y Cierre | 15h | 15h | 20h | 10h | $1.980.000 |
| **Totales Proyecto** | **90h** | **105h** | **110h** | **45h** | **$11.290.000** |

# **Premisas y restricciones**

**11.1Premisas**

1. Se contará con acceso a los datos históricos de ODEPA en formato CSV publicados en su portal oficial.
2. Los sitios web de supermercados mantienen la información de precios accesible públicamente para el scraping.
3. Los usuarios finales (consumidores, pymes o instituciones) tienen acceso a internet y un navegador moderno.
4. El equipo de desarrollo dispondrá de conocimientos en Python, Django, React, Vite yTailwindCSS.
5. El proyecto se desarrollará en un entorno controlado (servidor local o institucional) antes de considerar su despliegue en nube.
6. El versionamiento y colaboración se gestionará a través de GitHub.
7. Se asumirá que la canasta básica estará definida y acotada a un conjunto específico de productos.

**11.2 Restricciones**

1. **Base de datos:** La primera versión se implementará únicamente en PostgreSQL, limitando la concurrencia y escalabilidad.
2. **Tiempo:** El desarrollo debe completarse dentro del plazo definido por el cronograma académico/profesional.
3. **Recursos humanos:** El equipo de desarrollo será reducido, lo que restringe el alcance de funcionalidades avanzadas.
4. **Seguridad:** El sistema no contará en esta fase con mecanismos avanzados de autenticación o cifrado; se implementará seguridad básica.
5. **Disponibilidad de fuentes:** Si ODEPA cambia el formato de publicación de datos o los supermercados modifican su estructura web, el scraping podría verse afectado.
6. **Alcance tecnológico:** No se consideran integraciones con sistemas externos (p. ej., ERP, sistemas de pago o logística).
7. **Presupuesto:** Uso restringido a herramientas open source y entornos gratuitos (PostgreSQL, Django, React, GitHub).

# **Riesgos iniciales de alto nivel**

Los Riegos de alto nivel del proyecto son los siguiente:

| **ID** | **Riesgo** | **Probabilidad** | **Impacto** | **Plan de Mitigación** |
| --- | --- | --- | --- | --- |
| R1 | Cambios en el formato de publicación de datos de ODEPA. | Media | Alto | Diseñar procesos de carga flexibles y documentados; validar periódicamente el formato. |
| R2 | Bloqueo o cambios en la estructura de los sitios de supermercados (scraping). | Alta | Alto | Implementar scraping modular y actualizable; usar múltiples fuentes como respaldo. |
| R3 | Limitaciones de escalabilidad de SQLite ante mayor volumen de datos. | Alta | Medio | Diseñar el modelo con ORM de Django para migrar fácilmente a PostgreSQL en el futuro. |
| R4 | Retrasos en el desarrollo por curva de aprendizaje de tecnologías nuevas (Vite, Tailwind). | Media | Medio | Planificar sesiones de capacitación interna y pruebas rápidas con prototipos. |
| R5 | Baja adopción por parte de usuarios finales si la interfaz no es intuitiva. | Media | Alto | Realizar pruebas de usabilidad tempranas; priorizar diseño responsive y simple. |
| R6 | Pérdida o corrupción de datos en procesos de scraping o carga. | Baja | Alto | Implementar validaciones y respaldos periódicos de la base de datos. |
| R7 | Limitaciones de tiempo para implementar todas las funcionalidades planificadas. | Alta | Alto | Definir un alcance mínimo viable (MVP) y dejar mejoras para fases futuras. |
| R8 | Falta de conectividad o problemas de red que impidan acceder a las fuentes de datos. | Media | Medio | Permitir carga manual de archivos CSV como respaldo; cachear datos previos. |
| R9 | Problemas de compatibilidad entre librerías de frontend y backend. | Baja | Medio | Usar versiones estables y documentadas; pruebas de integración continuas. |

# **Cronograma de hitos principales revisar**

Los hitos principales para revisar en todo el ciclo de vida del proyecto son los siguientes:

| **Hito** | **Descripción** | **Semana** | **Fase** |
| --- | --- | --- | --- |
| **Inicio de Capstone** | Inicio formal del proyecto, organización del equipo y definición de lineamientos. | S1 | Fase 1 |
| **Análisis del proyecto** | Revisión de requerimientos, identificación de objetivos y delimitación del alcance. | S2 | Fase 1 |
| **Definición del proyecto y diseño** | Elaboración del diseño inicial de la arquitectura y documentación base del sistema. | S3 | Fase 1 |
| **Análisis de datos** | Revisión y limpieza de fuentes (ODEPA + supermercados) para asegurar consistencia. | S4–S5 | Fase 1 |
| **Desarrollo Backend** | Construcción del backend en Django, definición de modelos y APIs REST. | S6–S9 | Fase 2 |
| **Desarrollo Frontend** | Implementación de la interfaz en React + Vite + TailwindCSS, conectada al backend. | S10–S13 | Fase 2 |
| **Integración de datos** | Conexión entre BD, scraping y dashboards; pruebas de integración completas. | S14–S16 | Fase 2 |
| **Gestión de proyecto** | Control, seguimiento de avances y ajustes en planificación según cronograma. | S15 | Fase 3 |
| **Presentación final** | Entrega de resultados, exposición de la plataforma y documentación. | S16–S18 | Fase 3 |

# **Lista de Interesados (Stakeholders)**

| **Interesado** | **Rol / Interés** | **Nivel de influencia** | **Expectativa** |
| --- | --- | --- | --- |
| **Equipo de Proyecto** | Jefe de Proyecto, Programadores (Back-End y Front-End), DBA | Alto | Ejecutar el desarrollo en tiempo y calidad definida. |
| **Usuarios finales** | Consumidores y pequeñas empresas | Medio | Acceder a dashboards claros y datos confiables sobre precios. |
| **Instituciones públicas** | Ejemplo: ODEPA, municipalidades | Medio | Obtener una herramienta que facilite la toma de decisiones con datos integrados. |
| **Supermercados** | Fuente de datos para el scraping | Bajo | Transparencia en la publicación de precios. |
| **Academia / Universidad** | Profesor guía, comisión evaluadora del capstone | Alto | Ver un producto funcional, con documentación y resultados demostrados. |
| **Área de soporte TI** | Encargados de servidores y despliegue (si aplica) | Bajo–Medio | Asegurar que la solución sea instalable y mantenible en el entorno asignado. |

# 

# **Requisitos de aprobación del proyecto**

1.-Cumplimiento de Alcance: El proyecto debe incluir todas las funcionalidades definidas en el alcance inicial: backend en Django, dashboards interactivos, módulo de scraping y comparador de canasta básica.

2.-Entrega de Documentación: Se debe presentar la documentación técnica (arquitectura, manual de instalación, manual de usuario) junto con el informe final.

3.-Validación Técnica: El sistema debe ser probado y validado en un entorno controlado, mostrando conexión con la base de datos, visualización de dashboards y funcionamiento del scraping.

4.-Aceptación Académica: El profesor guía y la comisión evaluadora deben aprobar la entrega en base a los criterios de la rúbrica del capstone.

5.-Presentación Final: El equipo debe exponer los resultados, demostrando el uso de la plataforma y los beneficios alcanzados.

6.-Calidad Mínima del Producto (MVP): Debe existir al menos una versión estable y navegable de la plataforma que cumpla con las funcionalidades básicas

# **Aprobaciones**

| **Patrocinador** | **Fecha** | **Firma** |
| --- | --- | --- |
| Felix Cifuentes | 20-11-2025 |  |
| Rodrigo Cea | 20-11-2025 |  |
| Jefe de Proyecto | 20-11-2025 |  |