

Ecotributario

Proyecto Fundamentos de Ingeniería de Software

Nicolas Parra, Sebastián Rodríguez, Sara Muñoz

1. Introducción

El desarrollo de soluciones tecnológicas orientadas a la sostenibilidad se ha convertido en una necesidad global. En este contexto, el proyecto "Ecotributario: Sistema de Incentivos Fiscales por Acciones Ambientales" surge como una iniciativa para fomentar prácticas ecológicas mediante beneficios tributarios. La creciente preocupación por el cambio climático y la degradación ambiental ha impulsado la creación de mecanismos que incentiven a individuos y empresas a adoptar comportamientos responsables con el medio ambiente.

2. Descripción del Proyecto

Nuestro grupo se enfocó en una propuesta que sea llamativa para las empresas que aún no estén vinculadas y para todas aquellas que ya están presentes, pero no quieren perder el beneficio dado. Para ello nuestro sistema busca fomentar la participación de las empresas grandes y pequeñas en actividades ambientales, ya sea el manejo de residuos plásticos, creación de espacios verdes y su mantenimiento o incluso las diferentes formas para la purificación de agua o mantenimiento de paramos. Todo esto con el fin de que las empresas reciban un descuento al momento de declarar impuestos como personas jurídicas, para lograr este objetivo nuestro proyecto permite el manejo de archivos para que alguien verificado los analice y les dé el visto bueno para los descuentos

3. Detalle del Proceso

3.1 ¿Qué se ha hecho hasta ahora?

Se ha implementado la metodología de trabajo Scrum, donde cada integrante del grupo se encarga de tareas específicas, de corta o larga duración. Hasta ahora se ha logrado:

- **README:** Documento descriptivo sobre la metodología y alcance del proyecto.
- **Issues:** Registro de actividades y tareas en el repositorio de GitHub, previamente acordadas en el grupo.
- **Develop Flow:** Flujo de trabajo basado en ramas, donde cada integrante realiza actividades en su respectiva rama para facilitar el control de versiones y actualizaciones.

3.2 ¿Cómo se está haciendo?

A través de la metodología Scrum, cada integrante del grupo asume tareas específicas para desarrollar el proyecto de forma eficiente. Entre estas tareas se incluyen la recopilación de requisitos, elaboración de casos de uso, creación de diagramas, y más. Esto permite mantener un flujo continuo de trabajo y realizar las actualizaciones necesarias de forma ágil.

3.3 Metodología Aplicada

Se ha aplicado la metodología **Scrum**, centrada en la iteración rápida, la colaboración continua y la adaptación al cambio. Esto es ideal para el proyecto, ya que a medida que aumentan las tareas, se pueden redistribuir eficientemente entre los integrantes, garantizando un avance constante.

4. Evidencia Visual

4.1 Capturas del Repositorio

- Imagen del repositorio en GitHub mostrando la estructura inicial.

File/Folder	Description	Time
.github	reestructuración del proyecto y eliminación de car...	19 hours ago
conf	reestructuración del proyecto y eliminación de car...	19 hours ago
docs	Añadir carpetas docs y src con archivos .gitkeep	19 minutes ago
jupyter	reestructuración del proyecto y eliminación de car...	19 hours ago
project-name	reestructuración del proyecto y eliminación de car...	19 hours ago
scripts	reestructuración del proyecto y eliminación de car...	19 hours ago
src	Añadir carpetas docs y src con archivos .gitkeep	19 minutes ago
temp	reestructuración del proyecto y eliminación de car...	19 hours ago
.gitignore	He eliminado archivos y carpetas que no usaremos	5 days ago
CHANGELOG.md	reestructuración del proyecto y eliminación de car...	19 hours ago
CONTRIBUTING.md	reestructuración del proyecto y eliminación de car...	19 hours ago
Dockerfile	reestructuración del proyecto y eliminación de car...	19 hours ago
LICENSE	He eliminado archivos y carpetas que no usaremos	5 days ago
Makefile	reestructuración del proyecto y eliminación de car...	19 hours ago
README.md	Update README.md	38 minutes ago
base.sql	ESQUEMA INICIAL BASE DE DATOS #6	17 hours ago
docker-compose.yml	reestructuración del proyecto y eliminación de car...	19 hours ago
project-nam	reestructuración del proyecto y eliminación de car...	19 hours ago

```
project-name/  
├── .github/  
│   ├── ISSUE_TEMPLATE/  
│   │   ├── bug_report.md  
│   │   ├── feature_request.md  
│   │   └── PULL_REQUEST_TEMPLATE.md  
│   ├── workflows/  
│   │   ├── ci.yml  
│   │   └── cd.yml  
├── conf/  
│   ├── config.yaml  
│   └── settings.json  
├── docs/  
│   ├── api/  
│   ├── architecture/  
│   └── user_guide/  
├── jupyter/  
│   ├── notebooks/  
│   │   ├── exploration.ipynb  
│   │   └── analysis.ipynb  
│   ├── datasets/  
│   │   ├── data1.csv  
│   │   └── data2.csv  
├── scripts/  
│   ├── setup.sh  
│   ├── deploy.sh  
│   └── test.sh  
├── src/  
│   ├── main/  
│   │   ├── java/ (o python/, etc. según el lenguaje)  
│   │   └── resources/  
│   ├── test/  
│   │   ├── java/ (o python/, etc. según el lenguaje)  
│   │   └── resources/  
├── temp/  
│   ├── temp_file.txt  
│   ├── temp_data/  
│   │   ├── temp1.tmp  
│   │   └── temp2.tmp  
├── .gitignore  
├── README.md  
├── LICENSE  
├── CHANGELOG.md  
├── CONTRIBUTING.md  
├── Dockerfile  
├── docker-compose.yml  
└── Makefile
```

se usó la distribución del boilerplate dada por el curso para el manejo de carpetas, cabe resaltar que para cada carpeta al inicio se usó un archivo .gitkeep para que git se obligue a rastrear dichas carpetas

Las ramas que usamos son las siguientes

```
Nicolás@LAPTOP-DEQLBCDM MINGW64 ~/fis_2025_g1/fis_2025_g1 (main|CHERRY-PICKING)
$ git branch
develop
feature/db-schema
feature/diagrams
feature/docs
feature/requisitos
* main
```

Acontinuacion se detallan las ramas del proyecto y su función

RAMA Main

- Solo se actualiza con cambios provenientes de features
- Se intenta no realizar desarrollos directos en esta rama

RAMA develop

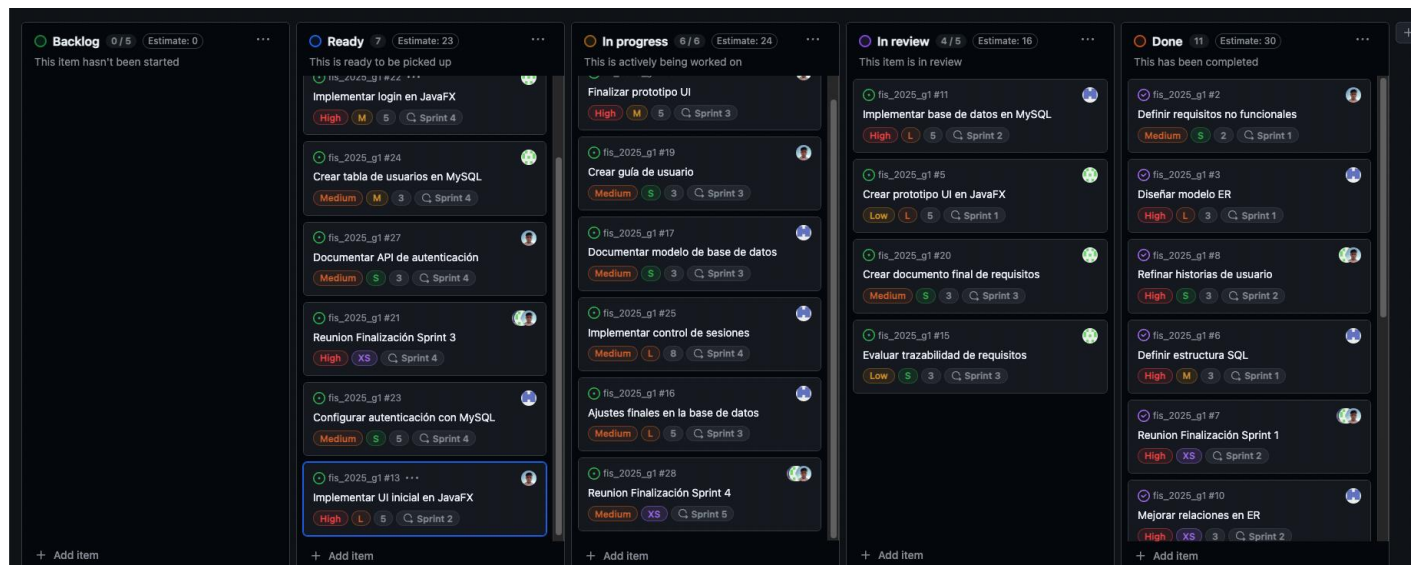
- Rama principal de desarrollo en una rama feature que se crea a partir de develop que se fusionara con develop una vez terminada

RAMA feature/*

- Cada nueva funcionalidad o modulo se desarrolla en una rama feature/* que se crea a partir de develop y se vuelve a fusionar con ella una vez terminada
- Feature/docs= documentación del proyecto
- Feature/db-schema creacion del esquema de la base de datos
- Feature/requisitos: implementacion y actualizacion de los requisitos del sistema
- Feature/diagrams diseño de diagramas de arquitectura y UML

4.2 Tablero Kanban

- Captura del tablero Kanban en GitHub Projects con las tareas iniciales categorizadas en (To do, In Progress, Done)



Actualmente, tenemos **7 tareas listas en Ready**, mientras que **6 están en progreso**, incluyendo la finalización del prototipo UI y la implementación del control de sesiones. Algunas tareas han tomado más tiempo del esperado, pero seguimos ajustando la planificación en nuestras **Daily Scrums**.

En **In Review**, hay **4 tareas** en validación, como la implementación de la base de datos en MySQL. Mientras tanto, en **Done**, ya completamos **11 tareas**, consolidando una base sólida para los siguientes sprints.

4.4 Gráficos del flujo de trabajo



Grafico1. Distribución de issues

Esta gráfica muestra la cantidad de issues en cada estado del flujo de trabajo: Ready, In Progress, In Review y Done. Se puede observar que la mayor cantidad de tareas se encuentra en la categoría Done, lo que indica un buen avance en el sprint y que muchas actividades ya han sido completadas. Por otro lado, los estados Ready e In Progress tienen una cantidad similar de issues, lo que sugiere que hay tareas en espera para ser tomadas y otras que se están ejecutando activamente. Esto refleja un flujo de trabajo continuo, donde las tareas avanzan progresivamente hacia su finalización.



Grafico 2. Issues Asignadas por Persona en el Sprint

En esta gráfica se representa la cantidad de issues asignadas a cada miembro del equipo, clasificadas por nivel de prioridad: High, Medium y Low. Se observa que algunos miembros tienen una mayor carga de issues de alta prioridad, mientras que otros tienen una distribución más equilibrada entre prioridades medias y bajas. Este análisis permite evaluar la equidad en la asignación de tareas dentro del equipo y detectar posibles sobrecargas de trabajo, asegurando que cada miembro tenga una carga manejable y alineada con su capacidad.



Grafico 3. Cantidad de Issues según Milestone y Prioridad

Esta gráfica muestra la cantidad de issues en diferentes hitos del proyecto (**Definición del Proyecto** e **Implementación Inicial**), categorizadas por nivel de prioridad. Se nota que en la fase de **Definición del Proyecto** hubo una mayor cantidad de issues, especialmente de prioridad alta, lo que indica que la planificación inicial requirió un esfuerzo significativo. En comparación, la fase de **Implementación Inicial** presenta menos issues en total, con una distribución más equilibrada entre prioridades. Esto sugiere que la mayor parte del trabajo se concentró en la fase de definición antes de pasar a la implementación.

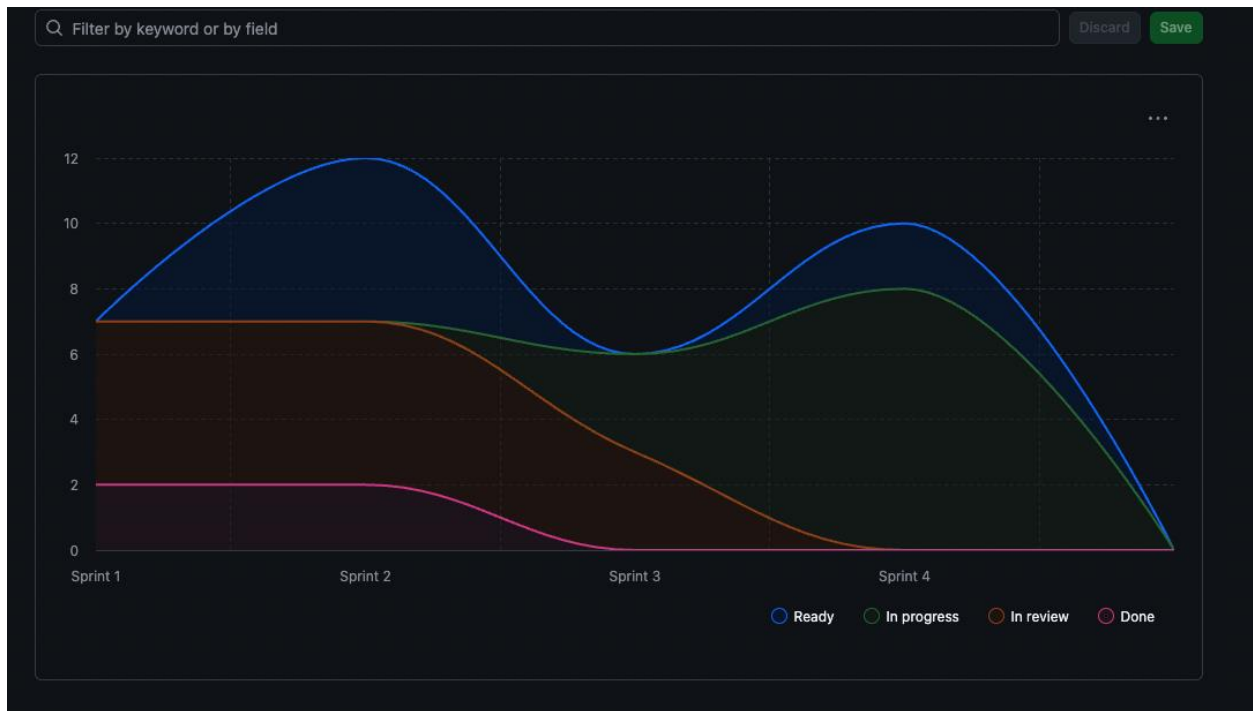


Grafico 4. Evolución de los Issues por Sprint

Esta gráfica representa la evolución de los issues a lo largo de los diferentes sprints, mostrando cómo las tareas han cambiado de estado en el tiempo. Se observa que, en los primeros sprints, hay un mayor número de issues en estado Ready y In Progress, lo que indica que muchas tareas estaban en cola o en desarrollo. Sin embargo, a medida que avanzan los sprints, el número de issues en Done aumenta, reflejando el progreso del equipo y la finalización de tareas. Este comportamiento es característico de metodologías ágiles, donde el trabajo avanza iterativamente hasta que la mayor parte de las tareas están completadas en los últimos sprints.

4.5 Issues

Diseñar modelo ER #3


ClosedTaskpuj-course/fis_2025_g1Public

sara-munoz-5 opened 3 weeks ago · edited by sara-munoz-5


Edits · ...


Descripción: Crear el diagrama de entidad-relación que define la estructura de la base de datos.
Criterios de aceptación: Modelo ER completo y validado con el equipo.

Create sub-issue


sara-munoz-5 added a commit that references this issue yesterday
Diagrama Entidad Relacion  Verified 46728b0

Assignees




 sara-munoz-5

Labels




Backend Diseño documentación


Type





Task

Projects



 KAMBAN_FIS_2025_G1

Status Done  


Priority High


Size L

Estimate 3

Iteration **Sprint 1**
Feb 18 - Feb 24

Milestone



 **Definición del Proyecto**


El issue "**Diseñar Modelo ER (#3)**" corresponde a la creación del diagrama de entidad-relación (ER) que define la estructura de la base de datos del proyecto. Su propósito es modelar las relaciones entre las entidades del sistema y establecer una base sólida para la implementación en MySQL. La tarea fue etiquetada bajo las categorías de backend, diseño y documentación.


Este issue tuvo una prioridad alta y fue clasificado con un tamaño grande dentro del proyecto KAMBAN_FIS_2025_G1. Se programó dentro del Sprint 1, con un plazo del 18 al 24 de febrero, y se incluyó en el milestone de Definición del Proyecto.


Para estimar la complejidad de la tarea, se utilizó el método Planning Poker, donde cada miembro del equipo asignó una puntuación basada en su percepción del esfuerzo requerido. Nicolás estimó un valor de 2, mientras que Sebastián y Sara asignaron un 3, dando como resultado un valor final de 3. Esto indica que la tarea presentaba una complejidad media, ya que aunque requería una planificación cuidadosa, era manejable dentro del sprint.

Con la finalización de este issue, el equipo cuenta con una estructura de base de datos bien definida, lo que permitirá avanzar con la implementación en los siguientes sprints.

Diseñar casos de uso #4

 Closed

 puj-course/fis_2025_g1 **Public**




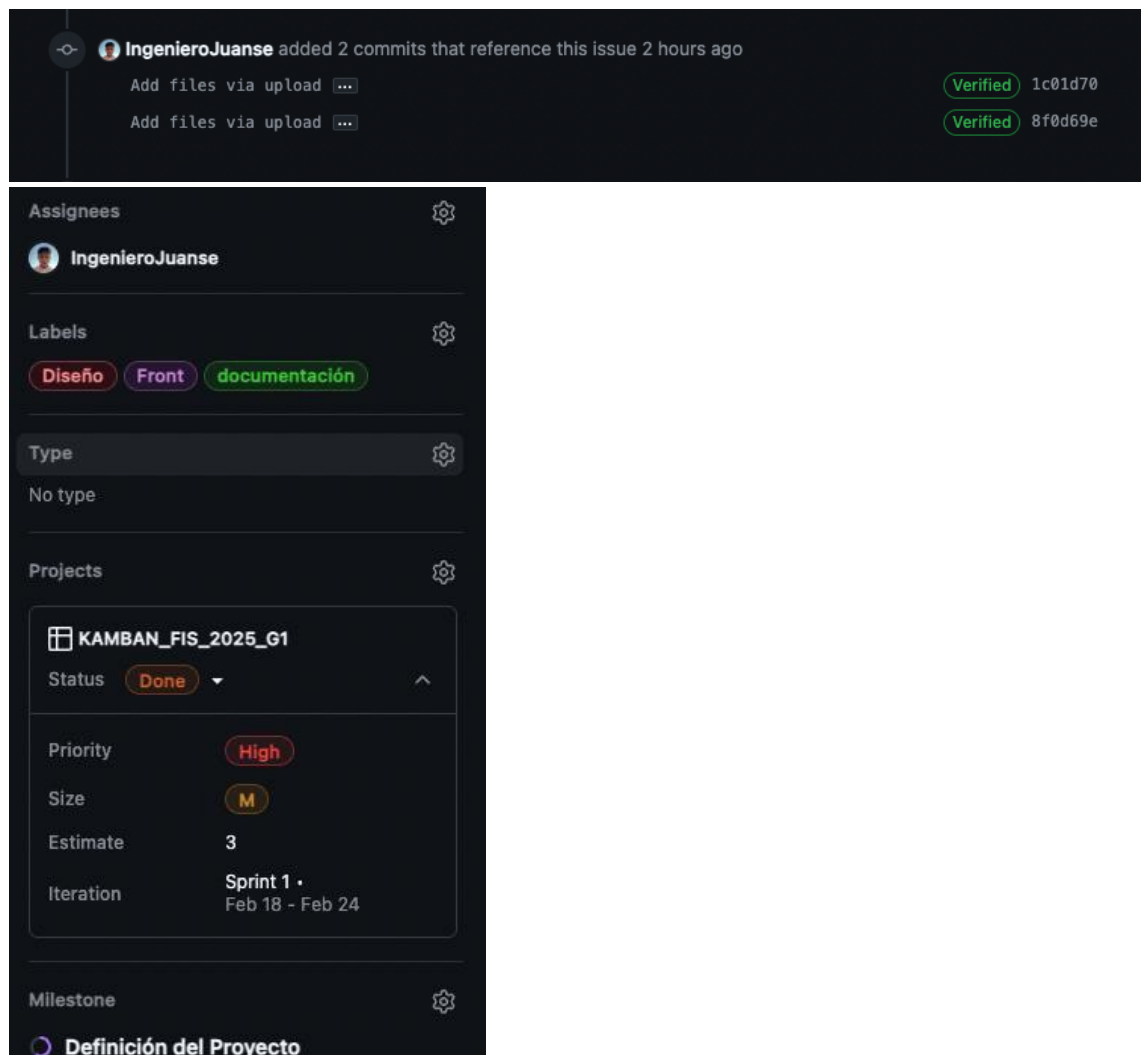
sara-munoz-5 opened 3 weeks ago · edited by sara-munoz-5

Edits ▾ ...

Descripción: Elaborar los casos de uso que describan la interacción de los usuarios con el sistema, detallando escenarios clave y actores involucrados.

Criterios de aceptación: Documento con al menos 5 casos de uso detallados con diagramas UML.

Create sub-issue ▾ 



El issue "Diseñar Casos de Uso (#4)" consistió en la documentación de los escenarios clave de interacción de los usuarios con el sistema, incluyendo al menos cinco casos de uso detallados con diagramas UML. Fue asignado a IngenieroJuanse y clasificado con prioridad alta dentro del Sprint 1 del proyecto KAMBAN_FIS_2025_G1. Se estimó con un valor de 3 en Planning Poker, lo que indica una complejidad media, ya que, si bien requirió un análisis estructurado, su alcance era manejable dentro del sprint. Finalmente, la tarea fue marcada como Done, y se registraron dos commits recientes donde se subieron los archivos con la documentación y diagramas verificados. Con la finalización de este issue, el equipo ahora cuenta con una base sólida para definir los procesos del sistema y guiar el desarrollo de las funcionalidades correspondientes.