
ReciclARG

**Sistema de Apoyo a los Punto Limpios
Software Architecture Document**

Version 1.0

Revision History

| Date | Version | Description | Author |
|------------|---------|-------------|--------|
| <dd/mm/yy> | <x.x> | <details> | <name> |
| | | | |
| | | | |
| | | | |

Table of Contents

| | | |
|-----|--|-------------------|
| 1. | Introduction | 4 |
| 1.1 | Purpose | 4 |
| 1.2 | Scope | 4 |
| 1.3 | Definitions, Acronyms, and Abbreviations | 4 |
| 1.4 | References | 4 |
| 1.5 | Overview | 4 |
| 2. | Use-Case Diagram | 4 |
| 3. | Structural Diagrams | 4 |
| 3.1 | Overview | 4 |
| 3.2 | Class Diagram | 4 |
| 3.3 | Packages Diagram | 5 |
| 3.4 | Deployment Diagram | 5 |
| 4. | Behavioral View | 5 |
| 4.1 | Overview | 5 |
| 4.2 | Activity Diagrams | 5 |
| 4.3 | Analysis Interaction Diagrams | 5 |
| 4.4 | Collaboration Diagrams | 5 |
| 4.5 | Sequence Diagrams | 5 |

Software Architecture Document

1. Introduction

1.1 Purpose

Este documento provee una descripción comprensiva del sistema, usando un número de diferentes vistas arquitecturales para describir distintos aspectos del sistema. Se dirige a capturar y transmitir las distintas decisiones que se han hecho en el sistema.

1.2 Scope

Este documento apunta a informar al PO y a los stakeholders una descripción del sistema y algunas de sus funcionalidades.

[A brief description of what the Software Architecture Document applies to; what is affected or influenced by this document.]

1.3 Definitions, Acronyms, and Abbreviations

PO: Product Owner.

SH: Steak Holder.

BD: Base de Datos.

1.4 References

Se referencian los siguientes documentos.

- Caso de uso: login (UC2)
- Caso de uso: registrar usuario (UC3)
- Caso de uso: registrar acopio (UC4)
- Caso de uso: ver estadística (UC5)

1.5 Overview

Este documento contiene una serie de diagramas, junto con una breve descripción de los distintos componentes del sistema. Se organiza en: Casos de uso, diagramas estructurales, y diagramas comportamentales.

2. Use-Case Diagram

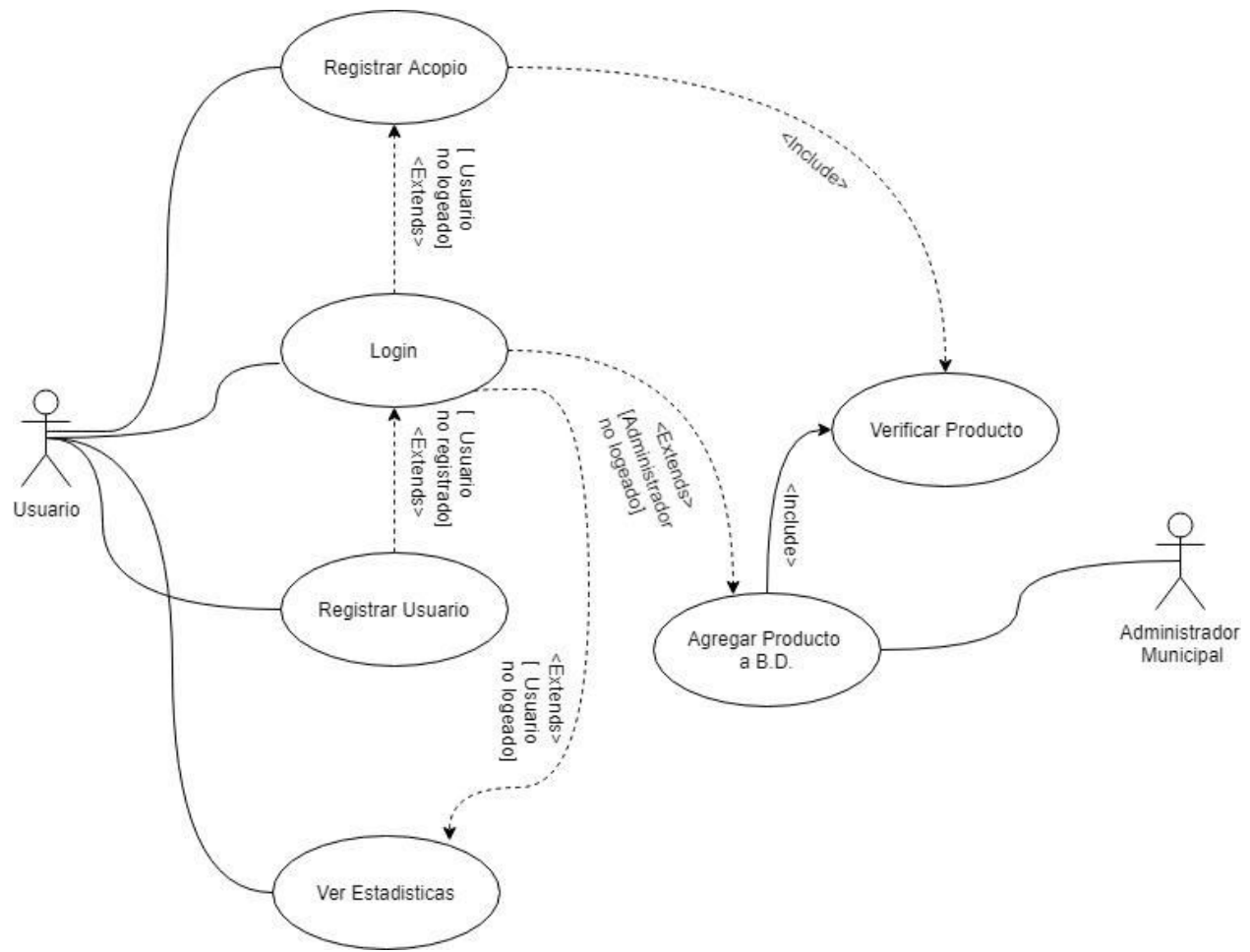


Figura 1: Diagrama de casos de uso

Los casos de uso representados en el diagrama son los que nos hemos enfocado en implementar en el primer sprint, las demás funcionalidades se han simulado o implementado parcialmente.

3. Structural Diagrams

3.1 Overview

Los diagramas estructurales del sistema se descomponen 2 partes. Por un lado se tiene el diagrama de clases, donde se representan todas las clases junto con sus atributos, funciones y relaciones. Y por otro el diagrama de paquetes, que ilustra cómo el equipo organizó estas clases para su implementación.

3.2 Class Diagram

En la figura 2 se contempla la división del sistema en clases, a continuación se describen las clases principales.

- Municipio: es un singleton y la base del sistema, es responsable de llevar el registro de los acopios de cada usuario para luego calcular sus estadísticas, también sería el responsable de la habilitación de puntos itinerantes y del envío de camiones pero estas 2 funciones no han sido implementadas en este sprint.
- User: esta clase contiene la información del usuario, su función es contener los acopios realizados por el mismo y proveer a otras clases el valor de sus atributos.
- UserApp: esta clase representa la aplicación que utilizará el usuario, su función proveer al vecino las funcionalidades básicas donde éste es el actor primario (UC2,UC3,UC4,UC5).
- UserManager: Contiene una colección de todos los usuarios del sistema, además se utiliza para la validación de credenciales.
- Producto: contiene los datos de un producto reciclable, su función es brindar información a la hora de calcular una estadística.
- ProductManager: contiene una colección de los productos que se consideran reciclables, además se utiliza para validar la existencia de un producto a la hora de registrar un acopio.

ReciclARG

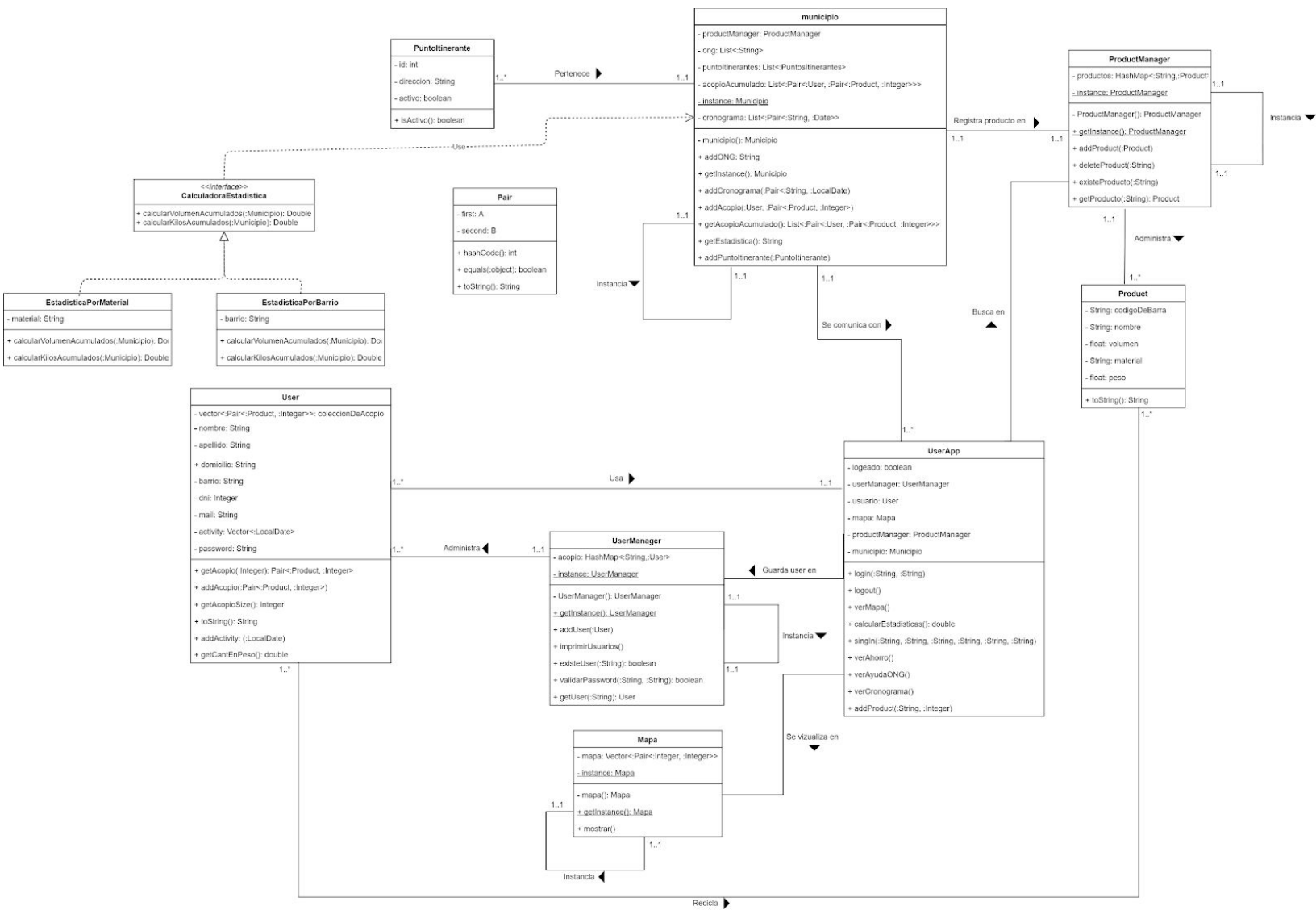


Figura 2: diagrama de clases

3.3 Packages Diagram

El sistema se descompone en 3 paquetes: Frontend, que se utilizara en la app del usuario. Backend, donde se almacena la información y se realizan las operaciones del sistema. Y Util, donde se incluyen clases de utilidades para la organización de datos.

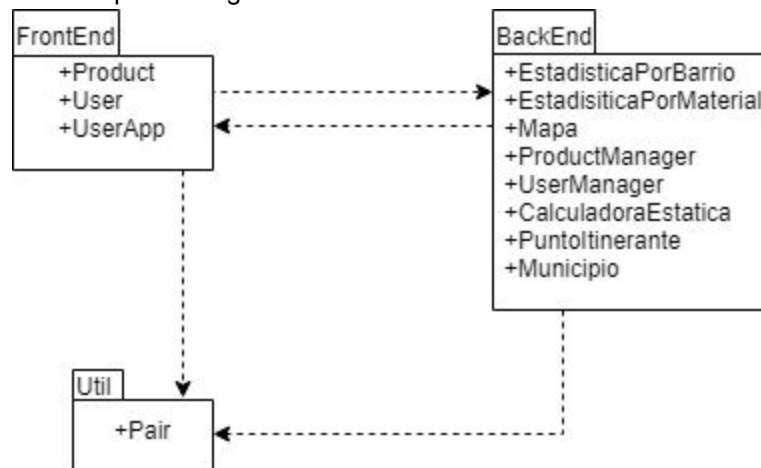


Figura 3: diagrama de paquetes.

4. Behavioral View

4.1 Overview

A continuación se presenta los diagramas de secuencia de las siguientes funcionalidades:

- Login
- Registrar acopio

4.2 Sequence Diagrams

En la figura 4 se describe el diagrama de secuencia que se ejecuta cuando se utiliza la funcionalidad descrita en UC2. Userapp recibe el mensaje y este consulta por la existencia del usuario, en caso de que exista a continuación se valida la password, si las credenciales son correctas se ingresa el usuario al sistema.

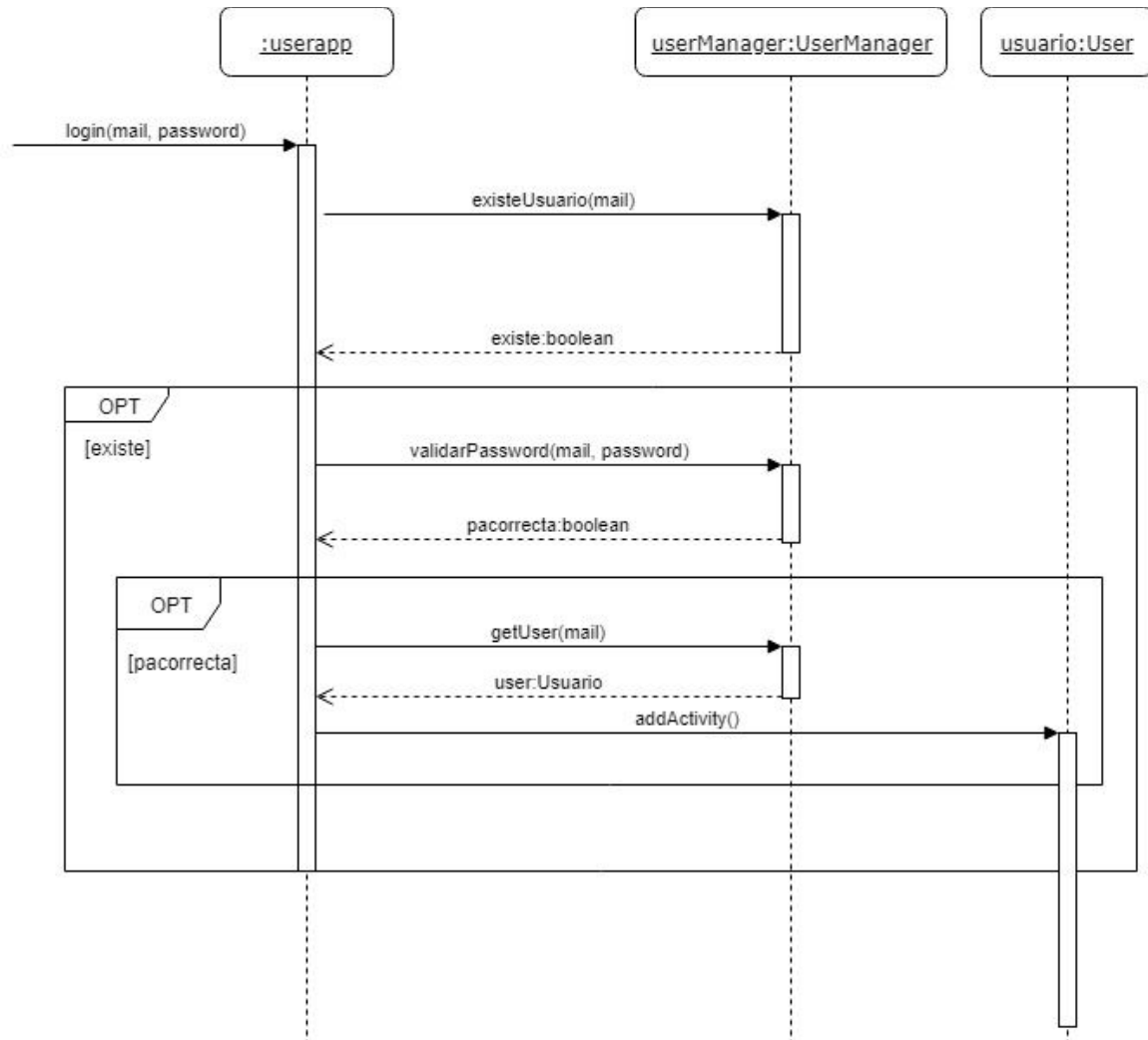


Figura 4: Diagrama de secuencia de funcionalidad login.

En la figura 5 se describe la secuencia que se ejecuta cuando se utiliza la funcionalidad descrita en UC4. En este caso Userapp recibe el mensaje de agregar un acopio y este verifica con ProductManager si el código de barras recibido existe en la BD, en caso de existir se obtiene el nombre del producto y luego se registra el acopio en User y en Municipio.

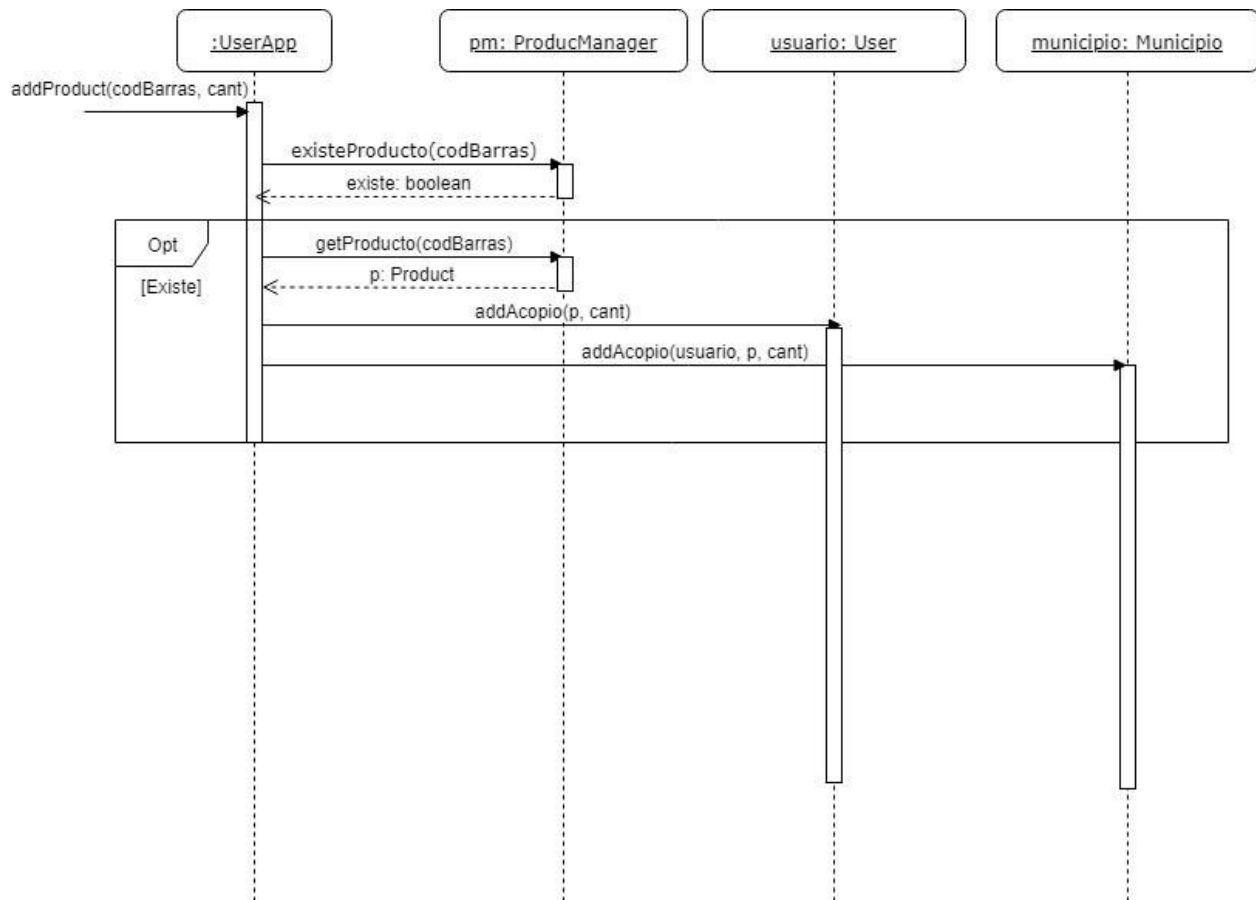


Figura 5: diagrama de secuencia de funcionalidad registrar acopio

[This subsection describes sequences of messages exchanges among roles that implement behavior of a system.]