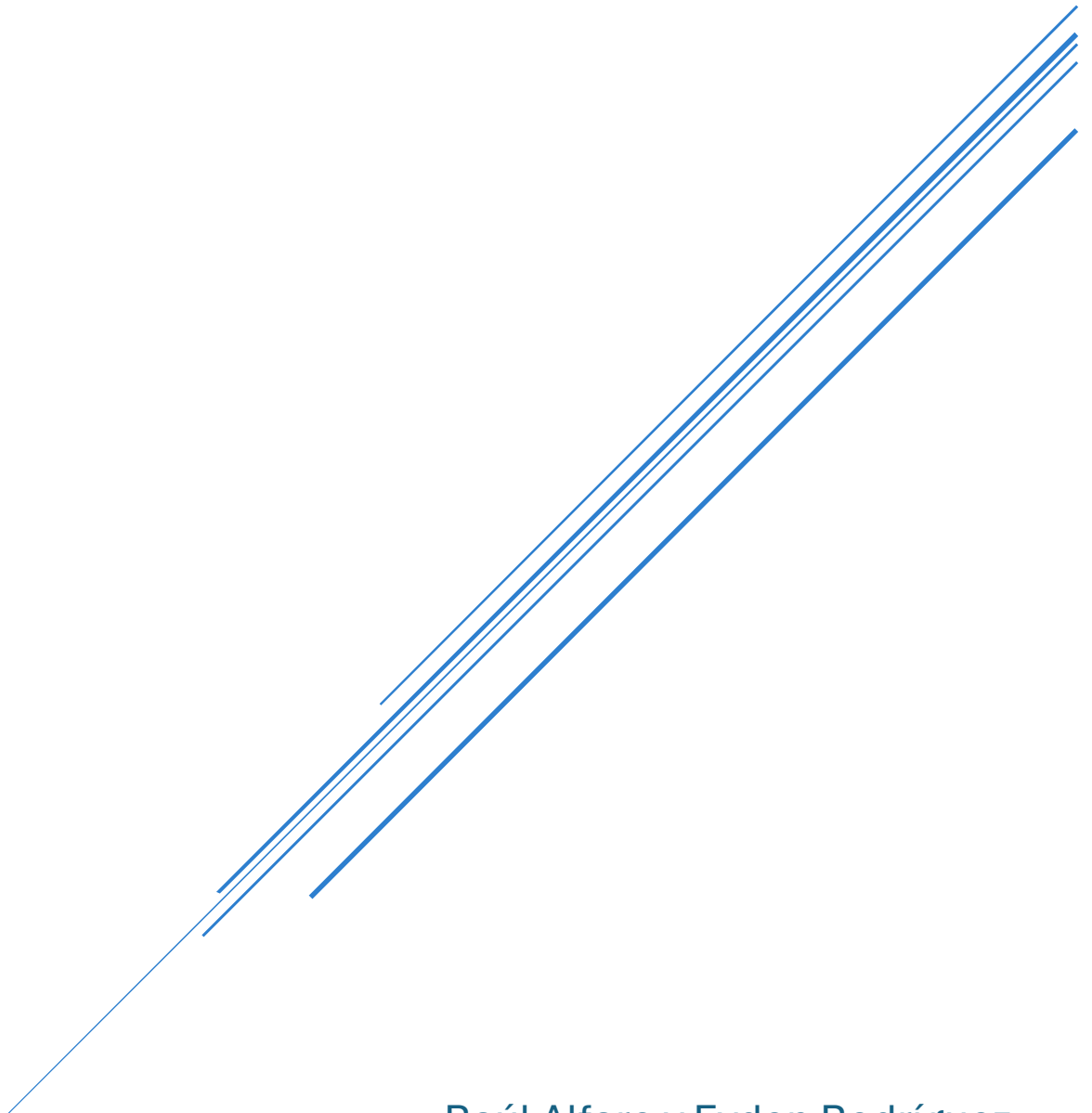


REQUERIMIENTOS DE SOFTWARE

Sistema de Gestión de Finca Agrícola



Raúl Alfaro y Eyden Rodríguez
Curso de Lenguajes de Programación

DESCRIPCIÓN GENERAL

Objetivo: Desarrollar una aplicación de escritorio de consola en Haskell para gestionar una finca agrícola, permitiendo el registro y consulta de parcelas, herramientas, trabajadores y cosechas, con generación de informes estadísticos.

Alcance:

- Registro y consulta de parcelas, herramientas y trabajadores.
- Gestión de cosechas (creación, cierre, modificación, cancelación).
- Consulta de disponibilidad de parcelas por fechas.
- Generación de reportes estadísticos.
- Persistencia opcional de datos (archivos o base de datos).

Restricciones:

- Datos en memoria (no persistentes, a menos que se implemente la opción extra).
- Validación de unicidad en códigos de herramientas y parcelas.
- Interfaz por consola (menús recursivos).

LISTA DE REQUERIMIENTOS

Requerimientos de Usuario

Funcionales

ID: RU-F001

Descripción: Los usuarios deben poder autenticarse ingresando su cédula, la cual debe estar predefinida en el sistema.

Importancia: Alta

Estimación Inicial: 1 Día

Estrategia de Implementación: Validar la cédula ingresada contra una lista predefinida de trabajadores almacenada en memoria.

ID: RU-F002

Descripción: Los usuarios deben poder cargar herramientas desde un archivo CSV con formato: código, nombre, descripción y tipo (manual, motorizada, automatizada).

Importancia: Media

Estimación Inicial: 2 Días

Estrategia de Implementación: Leer el archivo línea por línea, validar formato y unicidad del código, y almacenar en una lista en memoria.

ID: RU-F003

Descripción: Los usuarios deben poder registrar parcelas indicando nombre, zona, área, tipos de vegetal, precio por kilo y herramientas asignadas.

Importancia: Alta

Estimación Inicial: 3 Días

Estrategia de Implementación: Implementar un formulario interactivo que valide los datos y genere un ID único para cada parcela.

ID: RU-F004

Descripción: Los usuarios deben poder consultar parcelas por su identificador único, mostrando información general y herramientas asociadas.

Importancia: Media

Estimación Inicial: 1 Día

Estrategia de Implementación: Búsqueda en la lista de parcelas y mostrar detalles en formato legible.

ID: RU-F005

Descripción: Los usuarios deben poder registrar una cosecha indicando parcela, fechas, tipo de vegetal y cantidad esperada, validando disponibilidad.

Importancia: Alta

Estimación Inicial: 3 Días

Estrategia de Implementación: Validar superposición de fechas con otras cosechas y generar un ID único.

ID: RU-F006

Descripción: Los usuarios deben poder cerrar una cosecha registrando la cantidad real recolectada.

Importancia: Alta

Estimación Inicial: 2 Días

Estrategia de Implementación: Actualizar el estado de la cosecha y almacenar la cantidad real.

ID: RU-F007

Descripción: Los usuarios deben poder consultar disponibilidad de parcelas por rango de fechas.

Importancia: Media

Estimación Inicial: 2 Días

Estrategia de Implementación: Filtrar parcelas no ocupadas en el rango solicitado y mostrar resultados agrupados.

No Funcionales

ID: RU-NF001

Descripción: La interfaz debe ser intuitiva, con menús recursivos y opciones numeradas.

Importancia: Alta

Estimación Inicial: 1 Día

Estrategia de Implementación: Diseñar un menú principal con submenús y validación de entradas.

ID: RU-NF002

Descripción: El sistema debe mostrar mensajes de error descriptivos (ej.: "Cédula no registrada").

Importancia: Media

Estimación Inicial: 1 Día

Estrategia de Implementación: Implementar funciones de validación con retroalimentación clara al usuario.

Requerimientos de Sistema

Funcionales

ID: RS-F001

Descripción: El sistema debe validar disponibilidad de parcelas antes de registrar una cosecha.

Importancia: Alta

Estimación Inicial: 2 Días

Estrategia de Implementación: Comparar fechas de cosechas existentes con las nuevas solicitudes.

ID: RS-F002

Descripción: El sistema debe generar IDs únicos para parcelas y cosechas.

Importancia: Alta

Estimación Inicial: 1 Día

Estrategia de Implementación: Usar contadores o UUIDs básicos en memoria.

ID: RS-F003

Descripción: El sistema debe evitar duplicados en códigos de herramientas y parcelas.

Importancia: Media

Estimación Inicial: 1 Día

Estrategia de Implementación: Validar unicidad al agregar nuevos ítems.

ID: RS-F004

Descripción: El sistema debe calcular estadísticas de cosechas (top 3, trabajador más activo, etc.).

Importancia: Media

Estimación Inicial: 3 Días

Estrategia de Implementación: Implementar funciones de agregación y ordenamiento sobre los datos.

No Funcionales**ID: RS-NF001**

Descripción: El sistema debe manejar estructuras en memoria (sin persistencia, a menos que se implemente la opción extra).

Importancia: Alta

Estimación Inicial: 0 Días (implícito)

Estrategia de Implementación: Usar listas y estructuras de datos en Haskell.

ID: RS-NF002

Descripción: El sistema debe ser eficiente en consultas, incluso con grandes volúmenes de datos simulados.

Importancia: Baja

Estimación Inicial: 1 Día

Estrategia de Implementación: Optimizar algoritmos de búsqueda y filtrado.

Requerimientos de Software

Funcionales

ID: RW-F001

Descripción: La aplicación debe implementarse en Haskell, siguiendo paradigma funcional (inmutabilidad, funciones puras).

Importancia: Alta

Estimación Inicial: N/A (base del proyecto)

Estrategia de Implementación: Usar tipos de datos algebraicos y evitar efectos secundarios.

ID: RW-F002

Descripción: Debe usarse entrada/salida por consola con manejo recursivo de menús.

Importancia: Alta

Estimación Inicial: 2 Días

Estrategia de Implementación: Implementar loops recursivos para menús y validar entradas.

ID: RW-F003

Descripción: La carga de herramientas debe ser incremental (append) desde archivos CSV.

Importancia: Media

Estimación Inicial: 1 Día

Estrategia de Implementación: Leer archivos y concatenar datos a la lista existente.

No Funcionales**ID: RW-NF001**

Descripción: El código debe ser modular, con separación clara entre los archivos de código fuente y los datos de almacenamiento.

Importancia: Alta

Estimación Inicial: 2 Días

Estrategia de Implementación: Dividir en módulos

ID: RW-NF002

Descripción: Debe incluir validaciones para evitar errores en tipos de datos (ej.: fechas inválidas).

Importancia: Media

Estimación Inicial: 1 Día

Estrategia de Implementación: Usar tipos customizados y funciones de validación.