+++ date = '2025-02-21T10:20:50-08:00' draft = false title = 'Practica3' +++

Practica 3

Primera Sesion

*Durante la primera sesion de esta practica instalamos lo necesario para utilizar **Haskell**.*

Segunda Sesion

*Ahora analizaremos un programa en **Haskell** para entenderlo de una mejor manera*

Descripción General

*Esta aplicación es una **ToDo App** desarrollada en Haskell, que permite gestionar una lista de tareas con operaciones básicas como crear, ver, actualizar y eliminar (CRUD). Está diseñada como un ejemplo práctico de cómo construir una aplicación web funcional usando Haskell y herramientas minimalistas.*

Estructura del Proyecto

- `Main.hs`: *Punto de entrada. Inicializa el servidor y define las rutas.*
- `Todo.hs`: *Define el modelo de datos y la lógica de negocio.*
- `Views.hs`: *Generación de HTML con BlazeHtml.*
- `style.css`: *Estilos básicos para la interfaz.*

Funcionamiento de la Aplicación

1. ***Inicio del servidor web:***
   Se ejecuta `Main.hs`, que levanta un servidor en `http://localhost:3000`.

2. ***Modelo de datos (`Todo`)***:

```haskell
data Todo = Todo {
  todoId :: Int,
  title :: Text,
  done :: Bool
}
```

Cada tarea tiene un ID, un título y un estado de completado.

Rutas definidas con Scotty

- `GET /` → *Muestra la lista de tareas.*
- `POST /add` → *Agrega una nueva tarea.*
- `POST /toggle/:id` → *Cambia el estado de completado.*
- `POST /delete/:id` → *Elimina una tarea.*

Vistas generadas usando BlazeHtml

***HTML** generado desde funciones en Haskell, de manera segura y declarativa.*

Estilo visual

Utiliza un archivo CSS simple ubicado en `static/style.css`.