

HISTORIAS DE USUARIO

Presentado por:

Nicolas Quezada Mora - nquezada@unal.edu.co

Sharick Yelixa Torres Monroy - shtorres@unal.edu.co

Laura Sofia Vargas Rodriguez - lavargasro@unal.edu.co

Jeronimo Bermudez Hernandez- jebermudez@unal.edu.co

Profesor:

Oscar Eduardo Alvarez Rodriguez - oalvarezr@unal.edu.co

Viernes 31 de Enero



Universidad Nacional de Colombia

Facultad de Ingeniería

Departamento de Industrial y Sistemas

2025

1. Historias de usuario:

Historia de Usuario #4

Descripción conceptual

Módulo	<i>Módulo de control de tareas</i>
Descripción de la(s) funcionalidad(es) requerida(s):	<i>Se requiere que exista un menú donde estén todas las tareas, que se actualice dependiendo del estado de cada una.</i>

Backend

Caso de uso técnico Al recibir el input de teclado J, se abre el menú de tareas, donde se muestran las tareas actuales y su estado.
Entrada Input de teclado WASD Implementación <pre>public partial class TaskManager : Node { private List<TaskData> _tasks = new List<TaskData>(); private Control _taskMenu; private VBoxContainer _taskList; private bool _menuVisible = false; public override void _Ready() { _taskMenu = GetNode<Control>("TaskMenu"); _taskList = GetNode<VBoxContainer>("TaskMenu/TaskList"); _taskMenu.Visible = false; public override void _Process(double delta) { if (Input.IsActionJustPressed("open_task_menu")) { ToggleTaskMenu(); } } private void ToggleTaskMenu() { _menuVisible = !_menuVisible; _taskMenu.Visible = _menuVisible; if (_menuVisible) {</pre>

```

        UpdateTaskList();
    }
}

private void UpdateTaskList()
{
    foreach (Node child in _taskList.GetChildren())
    {
        child.QueueFree();
    }

    foreach (var task in _tasks)
    {
        var taskLabel = new Label();
        taskLabel.Text = $"{task.Title} - {task.Status}\n{task.Description}";
        _taskList.AddChild(taskLabel);
    }
}
}

```

Frontend

Se abre el menú de tareas, donde el usuario puede revisar el estado de las tareas que tiene pendientes.