

## **PROYECTO ADMINISTRADOR DE TAREAS**

### **Descripción breve**

Se requiere una aplicación de escritorio para administrar tareas sencillas, con una interfaz de usuario optimizada y validación de datos.

### **Requisitos de la pantalla**

Se debe poder ver, crear, editar y eliminar tareas con la siguiente información:

1. **Descripción:** Campo que describe la tarea.
2. **Usuario:** Indica a quién está asignada la tarea.
3. **Estado:** Indica el estado actual de la tarea.
4. **Prioridad:** Indica la importancia de la tarea.
5. **Fecha de Compromiso:** Fecha en la que debe estar lista la tarea.
6. **Notas:** Campo para guardar notas asociadas a la tarea.

### **Visualización de las tareas**

1. Mostrar todas las tareas existentes en una lista tipo tabla, ordenada por fecha de compromiso.
2. La vista/componente debe incluir filtros.

### **Acciones permitidas**

- Las tareas deben poder moverse en este flujo: **PENDIENTE > EN PROCESO > TERMINADA**
- Editar: Solo las tareas en estado **PENDIENTE** pueden ser editadas.
- Eliminar: Las tareas pueden ser eliminadas siempre que no estén en estado **EN PROCESO**.

**Requerimientos Técnicos Adicionales:**

La solución deberá estar cumplir con lo siguiente:

**1. Interfaz de Usuario:**

- Puede utilizar una librería externa para optimizar la interfaz (se recomienda usar DevExpress, aunque es opcional). Debería tener un diseño limpio y amigable.

**2. Capa de acceso a datos:**

- Por ahora, el sistema deberá utilizar **SQLite** y **Dapper** como tecnología de capa de datos. Pero en los próximos meses migraremos esta capa de datos a Entity Framework. **La migración debería ser fluida y sin muchos cambios en todo el sistema.**

**3. Documentación:**

- Incluir un archivo **README.md** detallado, con instrucciones sobre cómo configurar el entorno de desarrollo, ejecutar el proyecto y una explicación de las decisiones técnicas.
- Añadir una breve guía de estilo de código que describa las convenciones utilizadas.

**Entrega:**

- La solución debe implementarse en C# utilizando **WinForms** y seguir una **arquitectura ordenada y adecuada para aplicaciones de escritorio.**
- Siéntase libre de hacer suposiciones razonables, pero asegúrese de que estén documentadas.
- Le invitamos a que haga preguntas si necesita aclaraciones sobre algún aspecto del desafío.
- Se busca una solución acorde a los conocimientos de un Senior, por lo que esperamos una solución de alta calidad.
- Subir el código fuente a un repositorio basado en Git (GitHub, Bitbucket, GitLab) e invitar a la dirección [srojas@ubitec.cr](mailto:srojas@ubitec.cr).
- El tiempo de entrega es de 1 semana.

### Rúbrica de evaluación

Item	Valor	Obtenido	Observaciones
Cumplimiento de requerimientos del aplicativo	20 pts		
Diseño de la interface	15 pts		
Prácticas de código limpio	15 pts		
Arquitectura propuesta en la solución (separación de capas, apego a patrones básicos, código ordenado)	25 pts		
Diseño y estructura de BD propuesta	10 pts		
Defensa del código	10 pts		
Entrega a tiempo	5 pts		
TOTAL	100 pts		