

# Taller 1: Agenda.

Algoritmos y Estructuras de Datos II, DC, UBA.

2022

La fecha de finalización sugerida es dos semanas después de la presentación del taller1.

En este taller se debe implementar una agenda. El enunciado descripto a continuación replica el Ejercicio 14 de la guía de ejercicios de laboratorio de la clase de definición de clases. Esta guía contiene sugerencias para la implementación de las partes que conforman la implementación a entregar.

La agenda debe iniciarse con la fecha del día actual. La fecha es referente al año corriente, por lo que debe definir día y mes. En una agenda deben poder agregarse recordatorios. Los recordatorios son asignados a un cierto horario en una fecha del año y contienen un mensaje describiendo que ha de recordarse. Una agenda debe poder presentar los recordatorios para el día actual. Finalmente, se debe poder incrementar el día en el sistema. Se debe cumplir la siguiente interfaz:

```
class Agenda {
public:
    Agenda(Fecha fecha_inicial);
    void agregar_recordatorio(Recordatorio rec);
    void incrementar_dia();
    list<Recordatorio> recordatorios_de_hoy();
    Fecha hoy();
};
```

Para conocer los recordatorios del día se espera que se sobrescriba el operador << y se impriman los recordatorios del día con el siguiente formato:

```
Agenda a(Fecha(5, 10));
a.agregar_recordatorio(Recordatorio(Fecha(5, 10), Horario(9, 0), "Clase Algo2"));
a.agregar_recordatorio(Recordatorio(Fecha(5, 10), Horario(11, 0), "Labo Algo2"));
cout << a << endl;
// 10/5
// =====
// Clase Algo2 @ 10/5 9:0
// Labo Algo2 @ 10/5 11:0
```

Notar que se imprime primero la fecha actual, luego un separador compuesto por el símbolo igual repetido cinco veces ("=====") y finalmente un recordatorio por línea. Los recordatorios del día deben imprimirse ordenados en orden cronológico (creciente por horario).

## Entrega

No vamos a exigir la entrega de este taller, no obstante recomendamos pasar todos los tests antes de continuar con el taller2.