

# Alumnos

Almacenamos la siguiente información de los alumnos de una facultad prestigiosa

- nombre
- dedicación, medida en horas de estudio
- materias aprobadas
- conceptos que maneja correctamente, ordenados de más recientemente adquiridos a menos recientes.

De las materias nos interesa su nombre y cuál es el efecto que produce sobre el alumno cuando la aprueba.

- "Paradigmas" es una materia que al aprobarla:
  - el alumno incorpora los conceptos "polimorfismo" y "orden superior"
  - su dedicación se incrementa en 100 horas de estudio
- La materia "Sistemas Operativos" impacta distinto al alumno dependiendo de la cantidad de reentregas de TP que se haga en la cursada para aprobarla, entonces:
  - la dedicación del alumno se incrementa en 1.000 horas de estudio por cada reentrega de TP
  - al nombre del alumno se le agrega el prefijo "Excelentísimo@ "
- La materia "Recursividad a full" te agrega una cantidad infinita de conceptos: "Recursividad 1", "Recursividad 2", "Recursividad 3", ...
- Por último en la materia "Desastre 2" se explica mal polimorfismo, por lo que si el alumno manejaba el concepto "polimorfismo", lo desaprende.

Se pide

1. Modelar el alumno en general y mostrar un ejemplo.
2. a. Declarar el tipo Materia y definir funciones para crear las 4 materias descritas anteriormente.  
Tip: La función `show :: Show a => a -> String` puede ser de utilidad para una de ellas ;)  
b. Definir una función que haga que un alumno apruebe una materia, teniendo en cuenta que al hacerlo, la misma se incorpora a las materias aprobadas del alumno además de afectar al alumno de acuerdo a la materia que sea como se indicó anteriormente.  
Sabemos que vamos a necesitar modelar más materias en el futuro, con lo cual extender el programa en este sentido debería ser fácil.
3. Mostrar cómo se lograría que un alumno apruebe Paradigmas, Recursividad a full y Sistemas Operativos en la 3er reentrega de TP **sin armar una lista de materias**.
4. Modelar una cursada, que consiste en conjunto de materias a cursar. Mostrar cómo se lograría el efecto del punto 3 con este modelo considerando que hacer una cursada implica aprobar todas sus materias.
5. Queremos saber si una cursada es mejor que otra para un alumno en base a alguno de los siguientes criterios que el usuario elija:
  - cuando le deja más conceptos
  - cuando tiene menos horas de dedicación
  - cuando, tomando el concepto aprendido más reciente luego de la primera cursada, el mismo tiene más letras que el concepto aprendido más reciente de la segunda cursada.Teniendo esto en cuenta:
  - a. Definir la función que solucione este problema sin repetir lógica.
  - b. Explicitar el tipo de esta función.
  - c. ¿Será posible determinar si una cursada que incluye la materia "Recursividad a full" es mejor que otra para un alumno? Justificar conceptualmente analizando los distintos criterios de comparación.
6. Nos plantean un cambio en el modelo: las materias no se aprueban sólo por cursarlas, ya que cada una podría tener distintas condiciones que el alumno debe cumplir para aprobarla.  
En principio sabemos que Paradigmas requiere que el alumno haya adquirido previamente el concepto "parametrización" para poder aprobar la materia, y Recursividad a full que haya aprobado la materia llamada "Recursividad a full". La materia Desastre 2 no requiere nada para aprobarla y Sistemas Operativos (independientemente de quién sea el alumno), que la cantidad de reentregas sea menor a 5. Realizar los cambios que sean necesarios para soportar este requerimiento.