

[Página Principal](#) / [Mis cursos](#) / [Carreras de Grado](#) / [Ingeniería en Informática](#) / [Período Lectivo 2020](#) / [tecprog](#) / [General](#)  
/ [Recuperatorio Lógico 2022](#)

<b>Comenzado el</b>	Friday, 8 de July de 2022, 16:06
<b>Estado</b>	Finalizado
<b>Finalizado en</b>	Friday, 8 de July de 2022, 17:50
<b>Tiempo empleado</b>	1 hora 44 minutos
<b>Calificación</b>	Sin calificar aún

## Pregunta 1

Finalizado

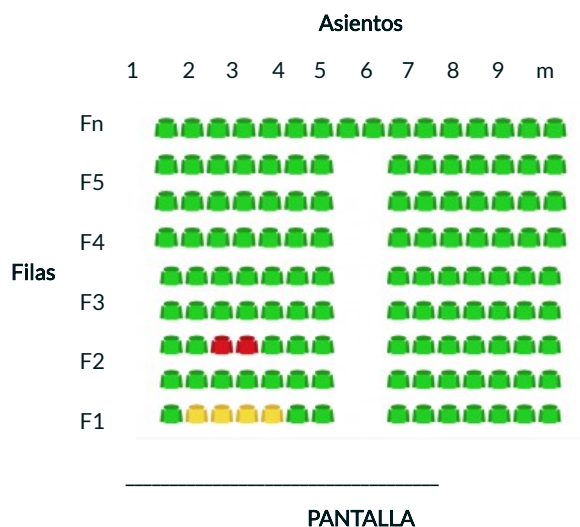
Puntúa como 100,00

Se desea programar un sistema que permita ver la disponibilidad de butacas en un cine en la que el usuario pueda elegir:

- Cantidad de butacas contiguas. Las butacas deberán estar adyacentes en la misma fila, sin butacas entre ellas
- Preferencia en evitar pasillos. Es decir, se puede elegir si dentro de las butacas elegidas puede haber butacas adyacentes al pasillo, o si se debe evitar toda butaca adyacente al pasillo.

Ofrecer diferentes posibilidades, considerando que para las fila 1, 2, 3, 8 y 9, la entrada cuesta \$800 y de la fila 4 a la 7 cuesta \$950 (menor precio aquel más cercano a la pantalla).

En el siguiente gráfico se puede observar el cine que se desea modelar. En las posiciones de las filas 1-8 y columnas 8-9 se encuentra el pasillo (este es el único pasillo que se considera tal). Se muestra una configuración de disponibilidad a modo de ejemplo. En verde se pueden ver las ubicaciones disponibles, en rojo los no disponibles y en amarillo las sugeridas según la preferencia. En el dibujo se ve que se ha sugerido cuatro ubicaciones que son las más baratas, al frente y sin preferencia por pasillo. Pero se deben ofrecer también otras posibilidades para la cantidad de butacas y preferencias de entrada del predicado principal.



Se recomienda resolver en el siguiente orden:

- Definir los hechos.
- Responder si hay lugar para esa cantidad de butacas contiguas, sin aclarar qué butacas se eligen, y sin tener en cuenta la preferencia por pasillo.
- Responder si hay butacas permitiendo evitar los pasillos si se lo desea, sin dar las ubicaciones de butacas elegidas.
- Finalmente, responder todas las posibilidades de butacas que tienen a disposición según la pregunta objetivo y el precio total de las entradas, por cada butaca ocupada informar su número de fila y columna, teniendo en cuenta la preferencia de pasillo.

## EJEMPLO:

cinemark(4, 'no\_pasillo', Butacas).

Butacas = [[fila1, butaca2], [fila1, butaca3], [fila1, butaca4], [fila1, butaca5], 4000];

Butacas = [[fila4, butaca2], [fila4, butaca3], [fila4, butaca4], [fila4, butaca5], 4750];

etc

IMPORTANTE: No utilice predicados predefinidos.

butacas(F,C):- F =< 8, C =< 16.

ocupadas([3,3],[3,4]).

pasillo(F,C):- F =< 1, F =\= 9, C = 7.

pasillo(F,C):- F = < 1, F = \= 9, C=10.

precio1([1,2,3,8,9]).

precio2([4,5,6,7]).

miembro(X,[X|\_]):-!.

miembro(X,[\_|R]):-miembro(X,R).

encontrar\_butaca\_no\_pasillo(Cant,Cant,\_,\_,[]):-!. % corta si el contador es igual a las butacas buscadas es el caso correcto

encontrar\_butaca\_no\_pasillo(\_,\_,\_,16,\_,\_):-!. %corta si la columna llego a su fin

encontrar\_butaca\_no\_pasillo(Cant,Cont,F,C,0,[["fila",F,"butaca",C]]Resul):-not(ocupadas(F,C)),butacas(F,C),not(pasillo(F,C)),C2 is C+1, Cont2 is Cont + 1, encontrar\_butaca\_no\_pasillo(Cant,Cont2,F,C2,1,Resul),!. %Encontramos butaca, activamos la bandera e incrementamos el contador

encontrar\_butaca\_no\_pasillo(Cant,Cont,F,C,1,[["fila",F,"butaca",C]]Resul):- not(ocupadas(F,C)),butacas(F,C),not(pasillo(F,C)),C2 is C+1, Cont2 is Cont + 1, encontrar\_butaca\_no\_pasillo(Cant,Cont2,F,C2,1,Resul),!. %si es butaca y no esta en el pasillo Y no esta ocupada seguimos e incrementamos el contador

encontrar\_butaca\_no\_pasillo(Cant,\_,F,C,1,[["fila",F,"butaca",C]]Resul):- butacas(F,C),pasillo(F,C),C2 is C+1,

encontrar\_butaca\_no\_pasillo(Cant,0,F,C2,0,Resul),!. %si esta en el pasillo cortamos la bandera y reiniciamos el cont

encontrar\_butaca\_no\_pasillo(Cant,\_,F,C,1,Resul):- butacas(F,C),ocupadas(F,C),encontrar\_butaca\_no\_pasillo(Cant,0,F,C,0,Resul). %si estan ocupadas cortamos la bandera y no concatenamos y reiniciamos el cont

%encontrar\_butaca\_no\_pasillo(4,0,1,1,0,R).

%encontrar\_butaca\_no\_pasillo(3,0,3,1,0,R).

%para el pasillo es lo mismo que sin pasillo pero una vez devuelta la lista debemos chequear si la ultima esta en el pasillo

encontrar\_butaca\_pasillo(Cant,Cant,\_,\_,[]):-!. % corta si el contador es igual a las butacas buscadas es el caso correcto

encontrar\_butaca\_pasillo(\_,\_,\_,16,\_,\_):-!. %corta si la columna llego a su fin

encontrar\_butaca\_pasillo(Cant,Cont,F,C,0,[["fila",F,"butaca",C]]Resul):-not(ocupadas(F,C)),butacas(F,C),C2 is C+1, Cont2 is Cont + 1, encontrar\_butaca\_no\_pasillo(Cant,Cont2,F,C2,1,Resul),!. %Encontramos butaca, activamos la bandera e incrementamos el contador

encontrar\_butaca\_pasillo(Cant,Cont,F,C,1,[["fila",F,"butaca",C]]Resul):- not(ocupadas(F,C)),butacas(F,C),C2 is C+1, Cont2 is Cont + 1, encontrar\_butaca\_no\_pasillo(Cant,Cont2,F,C2,1,Resul),!. %si es butaca Y no esta ocupada seguimos e incrementamos el contador

encontrar\_butaca\_pasillo(Cant,Cont,F,C,1,[["fila",F,"butaca",C]]Resul):-butacas(F,C),not(ocupadas(F,C)),C2 is C+1,Cont is Cant-1,Cont2 is Cont + 1,pasillo(F,C),encontrar\_butaca\_pasillo(Cant,Cont2,F,C2,1,Resul),!. %si el contador es igual a cantidad - 1 debemos chequear que la butaca este en el pasillo y si esta seguimos la recursion

encontrar\_butaca\_pasillo(Cant,Cont,F,C,1,[["fila",F,"butaca",C]]Resul):-butacas(F,C),not(ocupadas(F,C)),Cont is Cant-1,not(pasillo(F,C)),encontrar\_butaca\_pasillo(Cant,0,F,C,0,Resul),!. %si el contador es igual a cantidad - 1 debemos chequear que la butaca este en el pasillo y si no esta cortamos, reiniciamos contador y bandera

encontrar\_butaca\_pasillo(Cant,\_,F,C,1,Resul):- butacas(F,C),ocupadas(F,C),encontrar\_butaca\_no\_pasillo(Cant,0,F,C,0,Resul). %si estan ocupadas cortamos la bandera y no concatenamos y reiniciamos el cont

%encontrar\_butaca\_pasillo(1,0,1,1,0,R).

encontrar\_butaca2(\_,9,\_,\_,\_). %Corta cuando ya recorrio todas las filas

encontrar\_butaca2(Cant,F,C,Bandera,Lista):-encontrar\_butaca\_no\_pasillo(Cant,0,1,1,Lista), F2 is F+1

,encontrar\_butaca2(Cant,F2,C,Bandera,Lista). %va recorriendo todas las columnas llamando a encontrar\_butaca\_no\_pasillo y las filas llamandose recursivamente

encontrar\_butaca1(\_,9,\_,\_,\_). %Corta cuando ya recorrio todas las filas

```
encontrar_butaca1(Cant,F,C,Bandera,Lista):-encontrar_butaca_pasillo(Cant,0,1,1,Lista), F2 is F+1 ,encontrar_butaca1(Cant,F2,C,Bandera,Lista).
%va recorriendo todas las columnas llamando a encontrar_butaca_no_pasillo y las filas llamandose recursivamente
```

```
suma([],0):-!.
```

```
suma([X|Resto],Resul):-suma(Resto,Resul2), gasto(X,_,_), con_descuento(X,Suma), Resul is Resul2 + Suma,!.
```

```
suma(_|Resto,Resul):-suma(Resto,Resul).
```

```
precio_aux([_,F,_,_],800)😄 recio1(X),miembro(F,X),!.
```

```
precio_aux([_,F,_,_],950)😄 recio2(X),miembro(F,X).
```

```
%precio_aux([_,F,_,_],Resul).
```

```
precio([],0):-!.
```

```
precio([X|Resto],Resul)😄 recio(Resto,Resul2),precio_aux(X,Total), Resul is Total + Resul2,!.
```


```
precio(_|Resto,_)😄 recio(Resto,_).
```

```
%precio([[_1,_,_],[_2,_,_],[_3,_,_]], Resul).
```

```
encontrar_butaca(Cant,Cond,Lista):-Cond == 'no_pasillo',encontrar_butaca2(Cant,Cond,Lista).
```

```
encontrar_butaca(Cant,Cond,Lista):-Cond == 'pasillo',encontrar_butaca1(Cant,Cond,Lista).
```

```
cinemark(Cant,Cond,[Lista|Total]):-encontrar_butaca(Cant,Cond,Lista),precio(Lista,Total).
```

 [recuperatorioProLog.pl](#)

◀ [Recuperatorio Funcional 2022](#)

Ir a...

[Recuperatorio POO 2022](#) ▶