

Test Programación

Programación

- 1.- Escriba una función/método que determine la cantidad de 0's a la derecha de $n!$ (factorial)
- 2.- Escriba una función/método que dado un número entero, entregue su representación en palabras, Ej. 145 -> "ciento cuarenta y cinco"
- 3.- Considere un tablero de ajedrez de $N \times N$, realice un algoritmo que visite cada espacio del tablero, usando solamente los movimientos de un caballo. (Puntos extras si se visita cada espacio una sola vez)

Modelo de datos

- 1.- Un colegio necesita un sistema para administrar sus cursos. El sistema tiene que soportar que se le ingresen varios cursos. Cada curso tendrá un profesor a cargo y una serie de alumnos inscritos. Cada profesor, así como cada alumno puede estar en más de un curso. Además cada curso tendrá una cantidad no determinada de pruebas, y el sistema debe permitir ingresar la nota para cada alumno en cada prueba. Todas las pruebas valen lo mismo.

Escriba a continuación las tablas que utilizaría para resolver este problema con los campos y llaves de éstas. Intente hacer el sistema lo más robusto posible, pero sin incluir datos adicionales a los que se plantean acá.

- 2.- Escriba un Query que entregue la lista de alumnos para el curso "programación".
- 3.- Escriba un Query que calcule el promedio de notas de un alumno en un curso.
- 4.- Escriba un Query que entregue a los alumnos y el promedio que tiene en cada ramo.
- 5.- Escriba un Query que lista a todos los alumnos con más de un ramo con promedio rojo.
- 6.- Se tiene una tabla con información de jugadores de tenis: PLAYERS(Nombre, Pais, Ranking). Suponga que Ranking es un número de 1 a 100 que es distinto para cada jugador. Si la tabla en un momento dado tiene solo 20 tuplas, indique cuantas tuplas tiene la tabla que resulta de la siguiente consulta:

```
SELECT c1.Nombre, c2.Nombre
```

```
FROM PLAYERS c1, PLAYERS c2
WHERE c1.Ranking > c2.Ranking
a) 400
b) 190
c) 20
d) imposible saberlo
```

Diseño

1.- Si usted estuviera resolviendo el problema del colegio con programación orientada a objetos, defina que clases usaría, métodos y las variables de estas clases. Puede utilizar el lenguaje que más le acomode o bien pseudos código.

2.- Diseñe un mazo de cartas (orientado a objetos) con propiedades y métodos básicos que considere para ser utilizado en distintas aplicaciones que utilicen cartas.

3. Diseño código frontend

```
var citas = {
  lunes: [
    {nombre: 'Daniel', hora_inicio: '08:00', hora_termino: '09:00'},
    {nombre: 'Daniel', hora_inicio: '09:30', hora_termino: '11:00'},
    {nombre: 'Daniel', hora_inicio: '15:00', hora_termino: '16:00'},
    {nombre: 'Daniel', hora_inicio: '17:00', hora_termino: '19:30'}
  ],
  martes: [
    {nombre: 'Daniel', hora_inicio: '08:00', hora_termino: '09:00'},
    {nombre: 'Daniel', hora_inicio: '11:30', hora_termino: '12:00'},
    {nombre: 'Daniel', hora_inicio: '15:00', hora_termino: '16:00'},
    {nombre: 'Daniel', hora_inicio: '17:00', hora_termino: '19:30'}
  ],
  miercoles: [
    {nombre: 'Daniel', hora_inicio: '08:00', hora_termino: '09:00'},
    {nombre: 'Daniel', hora_inicio: '10:30', hora_termino: '12:00'},
    {nombre: 'Daniel', hora_inicio: '15:00', hora_termino: '16:00'},
    {nombre: 'Daniel', hora_inicio: '17:00', hora_termino: '19:30'}
  ],
  jueves: [
    {nombre: 'Daniel', hora_inicio: '08:00', hora_termino: '09:00'},
    {nombre: 'Daniel', hora_inicio: '09:30', hora_termino: '12:00'},
    {nombre: 'Daniel', hora_inicio: '15:00', hora_termino: '16:00'},
    {nombre: 'Daniel', hora_inicio: '17:00', hora_termino: '19:30'}
  ],
  viernes: [
    {nombre: 'Daniel', hora_inicio: '08:00', hora_termino: '09:00'},
    {nombre: 'Daniel', hora_inicio: '09:30', hora_termino: '12:00'},
    {nombre: 'Daniel', hora_inicio: '15:00', hora_termino: '16:00'},
    {nombre: 'Daniel', hora_inicio: '17:00', hora_termino: '19:30'}
  ],
}
```

Construya una función o clase en JS que recibiendo el anterior JSON por parámetro, permita renderizar una agenda semanal en html y con bloques de 30 minutos como la siguiente:

«	Lunes 09	Martes 10	Miercoles 11	Jueves 12	Viernes 13	Sabado 14	Domingo 15	»
08:00	+	+						08:00
08:20	+	 Peter More						08:20
08:40	+	+						08:40
09:00	+	 Guillermo						09:00
09:20	 Cecilia As	+		+				09:20
09:40		+		+				09:40
10:00		+		+				10:00
10:20		+		+				10:20
10:40		+		 JAVIERA CO				10:40
11:00		+		+				11:00
11:20		+		+				11:20
11:40		+		+				11:40
12:00		+		+				12:00

La agenda debe contener los distintos bloques y pintar con el nombre del paciente, las horas que están tomadas.

Consideraciones:

La agenda NO debe tener interacción solo dibujarse en la pantalla.

No utilizar tablas, sólo DIVS

La agenda debe tener un ancho de 960px y esta centrada en la pantalla