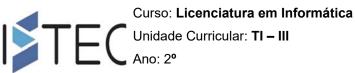
A CLASSIFICAÇÃO MÍNIMA NAS PARTES T/P e/ou LAB DO EXAME DEVERÁ SER 8.

Parte Teórica: 60% (12 Valores)

Grupo I (30%) Assinale a opção correta. Cada resposta errada desconta 1 certa, sendo que a menor cotação que se pode ter no grupo é 0.

- 1. Para que uma linguagem possa ser considerada "Object Oriented" deve:
 - a. Permitir o Encapsulamento, Herança, Polimorfismo
 - b. Permitir a Gestão de Eventos
 - c. Todas as anteriores
 - d. Nenhuma das anteriores
- 2. Entre as várias formas de definir estilos para conteúdos HTML, encontram-se a:
 - a. Inline
 - b. Embebida
 - c. Externa
 - d. Todas as opções anteriores
- 3. Podemos considerar como tipos de seletores para CSS:
 - a. Etiquetas ou "Tags" (utilizam-se keywords HTML)
 - b. Classes
 - c. ID
 - d. Todas as opções anteriores
- 4. Diz-se que ocorre um Postback quando:
 - a. Encontramos o corcunda de Notre-Dame (hunchback) numa página
 - b. A página submete o conteúdo de um form para outra página
 - c. A página submete o conteúdo de um form para si mesma
 - d. Nenhuma das opções anteriores
- 5. Numa página HTML5 padrão, o atributo ID pode ser utilizado para:
 - a. Identificar um objeto para manipular o seu conteúdo ou estilo em server-side
 - b. Enviar o conteúdo de um objeto do tipo Input (por exemplo) para ser lido server-side
 - c. Todas as anteriores
 - d. Nenhuma das anteriores
- 6. Entre as formas de enviar informação Controlador → View temos:
 - a. return View()
 - b. return View(objeto)
 - c. ViewBag e ViewData
 - d. Todas as opções anteriores



Tipo de Prova: Exame Tipo Data: 19-06-2023 Duração: 45 + 75 Minutos

Semestre: 2º

NOME: ______ N° _____

Grupo II (30%)

1. Considere o seguinte bloco de código:

```
public List<objetoRead> funcaoXPTO() {
   objeto doc = new objeto();
   List<objetoRead> contentor = new List<objetoRead>();
   DataTable dt = doc.funcaoXPTO();
   foreach (DataRow linha in dt.Rows) {
      objetoRead item = new objetoRead();
      item.uidObjeto = "" + linha["uidObjeto"];
      item.titulo = "" + linha["titulo"];
      item.outroAtributoQualquer = "" + linha["outroAtributoQualquer"];
      contentor.Add(item);
   }
   return contentor;
}
```

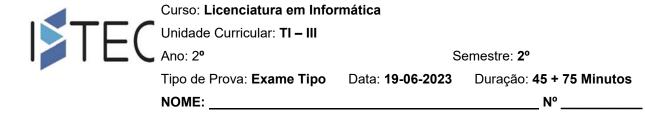
- a) Explique numa frase o objetivo desta função.
- b) Num contexto MVC onde consegue colocar esta rotina e qual seria o seu propósito (explique objetivamente e detalhadamente).
- c) Substitua os XPTO por nomes mais adequados
- 2. "Um controlador deve ser obrigatoriamente agnóstico em relação à plataforma onde está definido". Comente objetivamente a frase (V/F e porquê), e mencione outras entidades que conhece que devem ser agnósticas à plataforma onde estão instaladas.

Grupo III (40%)

Considere o seguinte cenário: Encomendam-lhe uma aplicação MVC para publicar no endereço https://BaterNoCeguinho.org

O objetivo é permitir a qualquer pessoa (mesmo anónima) escrever o que pensa sobre qualquer assunto sem ser penalizada por isso (tipo "Speaker's Corner" do Hyde Park) e poder comentar qualquer uma das opiniões escritas (ainda anónimo). Deve por isso:

- 1. **Elaborar** o parágrafo descritivo da aplicação, diagrama de classes, use-case e diagrama entidade-relação.
- Desenhar a "árvore" MVC que implementaria para produzir esta aplicação, incluindo eventuais classes principais e helpers (nota: SEM Código. Apenas <u>nomes</u> das entidades, tal como por exemplo, classe <u>ComentarioHelper</u> ou View <u>ListarComentarios</u>)



PROVA PRÁTICA: 40% (8 Valores)

Implemente o "BaterNoCeguinho" numa abordagem MVC. Deve:

- 1. Criar (65%):
 - a. Objetos necessários para gerir as opiniões (esqueça os comentários às opiniões)
 - b. Tabela necessária e respetivas instruções SQL
 - c. A estrutura MVC mínima e CORRETA para listar opiniões e inserir uma opinião. NÃO É NECESSÁRIO editar opiniões.
- 2. O projeto deve ficar totalmente funcional (35%)

Submeta no moodle a solução e os scripts da BD (Tabela e Instruções SQL)

Bom Trabalho