

ALP

PROVA DE AVALIAÇÃO
ATIVIDADE LETIVA

Tecnologias e Sistemas de Informação para a Web

CURSO

2023/2024	2024/06/06	14h00	120 minutos
-----------	------------	-------	-------------

ANO LETIVO

DATA

HORA

DURAÇÃO

Programação Orientada a Objetos	1º ano
---------------------------------	--------

UNIDADE CURRICULAR

ANO

Mário Pinto / Ricardo Queirós/Daniel Carneiro	Avaliação distribuída
---	-----------------------

DOCENTE

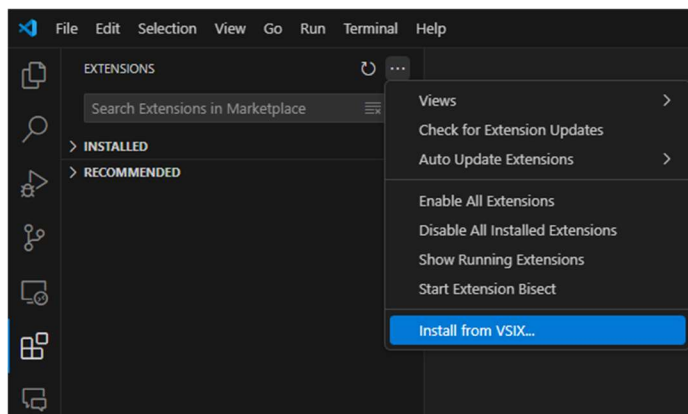
ÉPOCA

Observações:

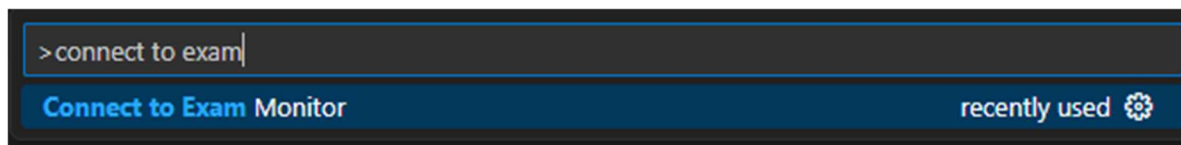
- O teste é individual.
- Descarregue a pasta **Teste2**. Mude o nome da pasta para o seu número de aluno
- **No final do teste submeta a pasta (compactada) no moodle, em objeto próprio de submissão**
- Podem consultar **APENAS** as seguintes referências externas: MDN, W3Schools, Moodle e GitHub (repo da UC e repo pessoal)
- **NÃO é permitido incluir código gerado por AI. É considerado plágio e resulta numa avaliação de 0**
- A avaliação do teste poderá implicar convocar os estudantes para uma sessão **de avaliação oral**, onde podem ter de explicar o código entregue
- Serão usadas ferramentas anti-plágio na avaliação dos testes

Instalação Exam Monitor

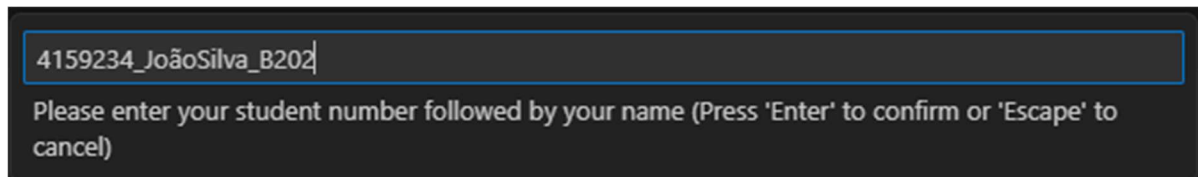
1. Aceda ao Moodle e na secção AVALIAÇÃO descarregue o ficheiro **exam-monitor-0.0.1.vsix**
2. Abra o VSCode e selecione o separador das extensões.
3. Clique nos “...” e selecione a opção **Install from VSIX...** e escolha o ficheiro que acabou de desc



4. Abra caixa de entrada de comandos usando **F1**, **CTRL+SHIFT+P** ou ir a **View>Command Palette...** e execute o comando “**Connect to Exam Monitor**”



5. Coloque a identificação no formato abaixo com o **número de aluno, nome e sala** , e pressionar **Enter**.

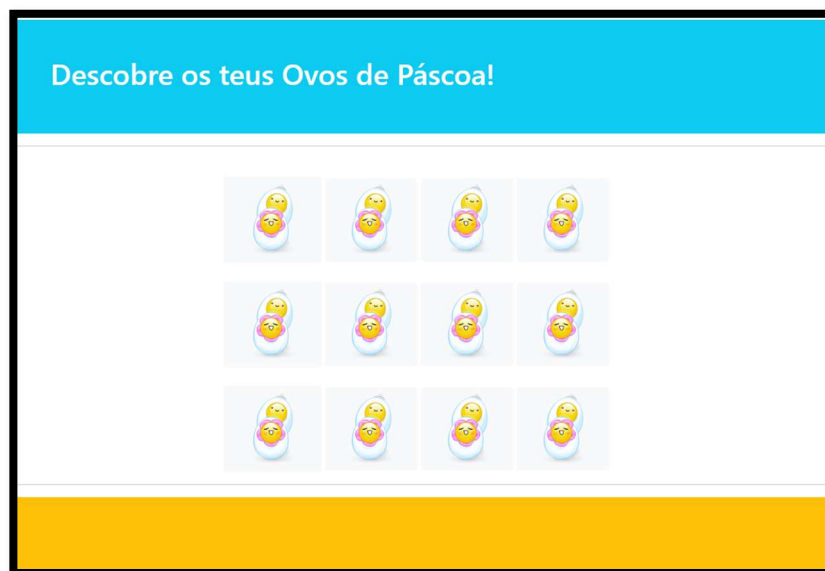


Aceder ao Moodle e descarregar o ficheiro materiais.zip com os ficheiros para as 2 partes do teste!

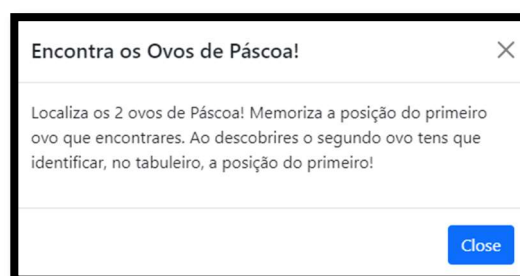
PARTE I

(50%)

1. Implemente uma pequena aplicação inspirada no jogo da memória, mas alusivo à descoberta de Ovos de Páscoa! A aplicação deve incluir:
 - 1.1 Um jumbotron com o formato abaixo apresentado
 - 1.2 Um tabuleiro com 12 posições (4x3) que inclua, em cada uma delas, a imagem *DiscoverEggs.png* (ficheiro disponível no Moodle);

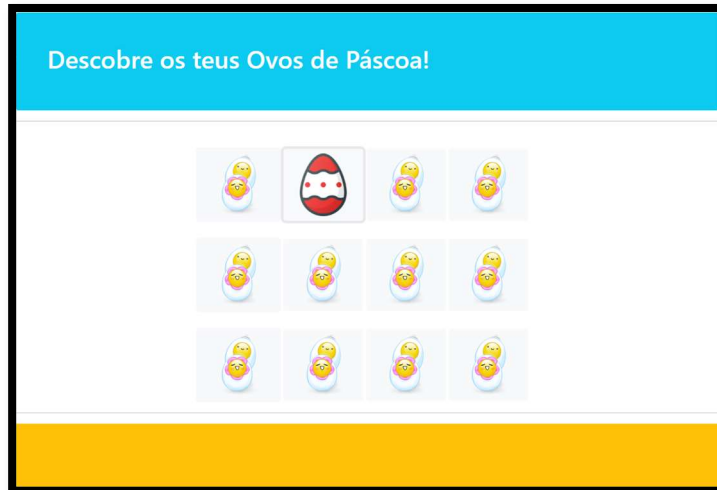


- 1.3 No arranque da aplicação deve surgir a seguinte modal:



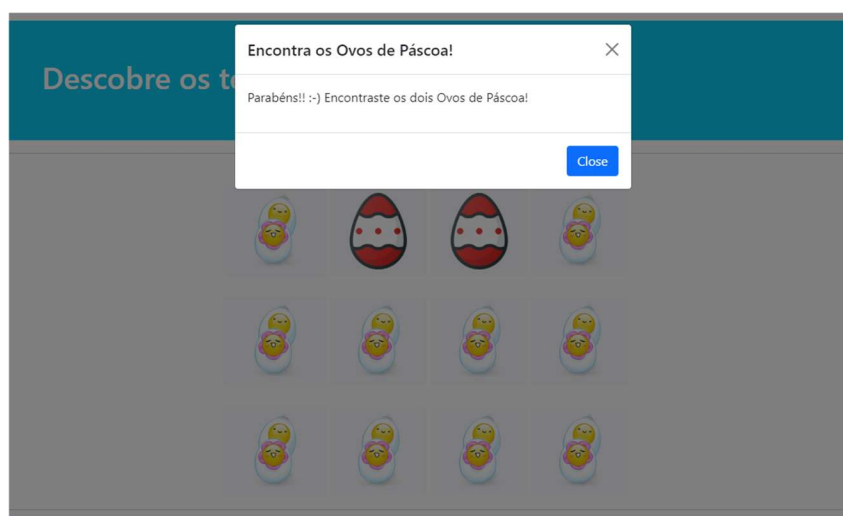
- 1.4 Os 2 ovos de Páscoa devem estar alocados, de forma aleatória, a duas posições do tabuleiro (não estando essas imagens, inicialmente, visíveis para o utilizador). Assim:
 - 1.4.1 Sortear as posições às quais os dois ovos de Páscoa (imagem *Ovo.png*) vão ser alocados;
 - 1.4.2 O utilizador vai clicando nos elementos do tabuleiro: quando clica num elemento que não contém o Ovo de Páscoa, nada acontece;

- 1.4.3 Quando clicar num elemento que contém um Ovo de Páscoa, essa imagem (*Ovo.png*) deve ser renderizada no respetivo elemento da DOM, ficando visível apenas durante 3 segundos (para o utilizador decorar a sua posição).



- 1.4.4 Passados os 3 segundos, deve voltar a associar a imagem *DiscoverEggs.png* ao elemento do DOM.
- 1.4.5 Quando o utilizador encontrar o segundo Ovo de Páscoa (igual ao primeiro), ele deve ser renderizado no respetivo elemento da DOM, ficando visível de forma permanente.
- 1.4.6 Em seguida o utilizador deve clicar novamente na posição em que encontrou o primeiro Ovo de Páscoa.

Caso acerte na posição, os dois Ovos devem ficar visíveis (imagem abaixo) e deve surgir novamente a modal inicial, mas agora com o texto *Parabéns! :-)* Encontraste os dois Ovos de Páscoa!



Se o utilizador não acertar na posição do ovo de Páscoa inicial, então deve surgir a mesma modal, mas agora com o texto *Que Memória Fraca! Não acertaste na posição do primeiro Ovo!*

PARTE II

(50%)

2. Desenvolva uma *app* chamada **T@kePill** para gerir os medicamentos que cada paciente está a tomar. A *app* deve ter as funcionalidades de adicionar medicamentos aos pacientes e de visualizar um *dashboard* com as tomas de medicação.

Gestão de Pacientes e Medicamentos

Adicionar Medicamento

João Silva

XponLag

8

1

3

Adicionar Medicamento

Dashboard - Tomas de Medicação

Nome do Paciente	Medicamento	Dose	Hora da Toma	Ação
João Silva	XponLag	1	05/06/2024, 00:00:00	TOMA
João Silva	XponLag	1	05/06/2024, 08:00:00	TOMA
João Silva	XponLag	1	05/06/2024, 16:00:00	TOMA
João Silva	XponLag	1	06/06/2024, 00:00:00	TOMA
João Silva	XponLag	1	06/06/2024, 08:00:00	TOMA
João Silva	XponLag	1	06/06/2024, 16:00:00	TOMA
João Silva	XponLag	1	07/06/2024, 00:00:00	TOMA
João Silva	XponLag	1	07/06/2024, 08:00:00	TOMA
Maria Santos	Britex	1	05/06/2024, 00:00:00	TOMA
Maria Santos	Britex	1	06/06/2024, 00:00:00	TOMA

Notas:

- Os ficheiros HTML e CSS não devem ser alterados e não devem ser criados outros ficheiros JS.

O que deve fazer?

- 2.1 Implemente a classe **Medicamento** através da definição do construtor (5%)
- 2.2 Implemente a classe **Paciente** através da definição do construtor e da definição do método **adicionarMedicamento** para guardar medicamento no array **medicamentos** do paciente (15%)
- 2.3 Implemente a lógica para adicionar medicamentos a pacientes (botão **Adicionar Medicamento**) (20%)
- 2.4 Atualize o dashboard para listar todas as tomas de medicação, distribuídas ao longo dos dias, considerando a quantidade total de comprimidos e a frequência diária (30%)
- 2.5 Adicione a funcionalidade ao botão **TOMA** que deve invocar método **tomarMedicamento** na classe **Paciente**. O método deve verificar se as tomas de determinado medicamento já foram todas feitas. Nesse caso, o medicamento deve ser removido do array de medicamentos (15%)
- 2.6 Adicione suporte à persistência de dados através da LocalStorage (15%)