

TAREFA UD07-03

1. Login


O enunciado proporciona o formulario que contén:

- Un campo de texto para o nome de usuario (username).
- Un campo de contrasinal (password).
- Un botón para enviar os datos.
- Un elemento para mostrar mensaxes de erro (<p id="message">).

O formulario xa está definido no ficheiro index.html

Os datos contemplados para acceder son os seguintes:

Usuario	Clave
admin	1234
usuario	1234

A mockup of a login form titled "Iniciar Sesión" is shown. It features a light gray background with a white border, set against a solid blue background. The form contains two input fields: "Usuario" and "Contrasinal", both with light gray borders. Below these fields is a blue button with the text "Entrar" in white. The title "Iniciar Sesión" is centered at the top in a bold, black font.

1.1 Login.js

Implementa un sistema de inicio de sesión utilizando **AJAX** en **JavaScript** que permita enviar os datos do formulario de login ao servidor e xestionar a resposta.

Requisitos:

1. Formulario de login:

- O formulario debe conter dous campos de entrada:
 - [username](#) (nome de usuario).
 - [password](#) (contrasinal).
- Un botón para enviar os datos.

2. Petición ao servidor:

- Ao enviar o formulario, realiza unha petición **POST** ao endpoint **api/login.php**.
- Envía os datos do formulario ([username](#) e [password](#)) no corpo da petición en formato **JSON**.

3. Resposta do servidor:

- O servidor devolverá unha resposta en formato **JSON** co seguinte formato:
 - { "success": true } se o login é correcto.
 - { "success": false } se o login é incorrecto.

4. Comportamento do script:

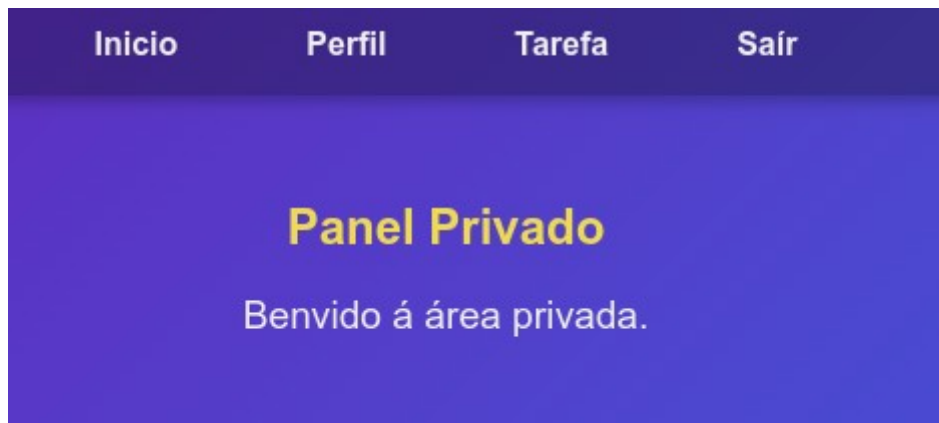
- Se o login é correcto ([success: true](#)), redirixe ao usuario á páxina `dashboard.html`.
- Se o login é incorrecto ([success: false](#)), mostra unha mensaxe de erro no elemento HTML co ID `message`.

Consideracións:

- Asegúrate de que o ficheiro [login.js](#) está correctamente enlazado no ficheiro HTML.
- O servidor debe estar configurado para recibir e procesar os datos enviados en formato JSON.
- Proba o script con diferentes credenciais para verificar o comportamento esperado.

2. dashboard

O ficheiro **dashboard.html** representa unha páxina de panel privado para usuarios autenticados.



O encabezado inclúe unha barra de navegación con ligazóns a diferentes seccións do panel:

- **Inicio:** Ligazón á páxina actual (`dashboard.html`).
- **Perfil:** Ligazón á páxina de perfil do usuario (`profile.html`).
- **Tarefa:** Ligazón a unha páxina de tarefas (`tarefa.html`).
- **Saír:** Un botón ou ligazón para finalizar a sesión do usuario.

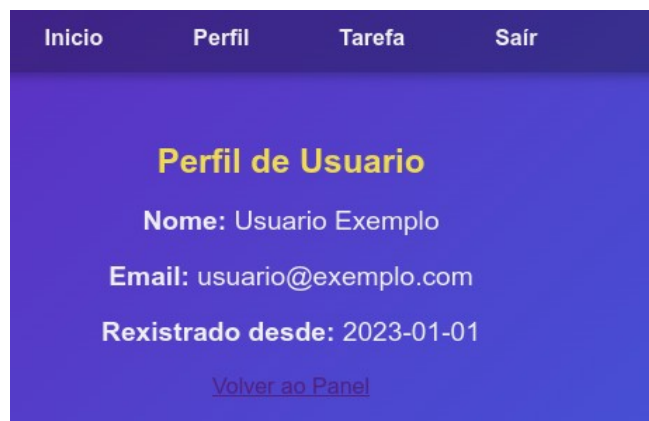
Ao final do corpo (`<body>`), inclúense varios ficheiros JavaScript para engadir funcionalidade á páxina:

- **profile.js:** Xestiona a funcionalidade relacionada coa páxina de perfil.
- **tarefa.js:** Xestiona a funcionalidade relacionada coas tarefas.
- **logout.js:** Xestiona o proceso de peche de sesión (logout).

2.1 Perfil (profile.js)

O script ten que realizar:

- Asociar un event listener á ligazón que apunta a profile.html e este realiza unha petición POST ao endpoint **api/profile.php** para obter os datos do perfil do usuario:
 - Realiza a petición empregando fetch a url indicada.
 - A petición realizase por “POST” e indicase no header que é unha aplicación JSON e inclúe as cookies na petición, necesarias para xestionar a sesión do usuario..
 - Envía un obxecto JSON co campo action para indicar ao servidor que debe devolver os datos do perfil.



O servidor devolve unha resposta en formato JSON, que o script procesa:

- response.ok: Verifica se a resposta do servidor foi exitosa (código HTTP 200).
- A resposta será como a seguinte en función do usuario:

```
{
  "role": "user",
  "name": "Usuario Exemplo",
  "email": "usuario@exemplo.com",
  "registered_since": "2023-01-01"
}
```

- **Logo terás que mostrar os datos do perfil:**
 - O script debe actualizar dinámicamente o contido da sección `<main>` para mostrar os datos do perfil do usuario.
 - O contido debe incluír:
 - Nome do usuario.
 - Email do usuario.
 - Data de rexistro.
 - Engade unha ligazón para volver ao panel (`dashboard.html`)
- **Xestión de erros:** Se ocorre un erro na petición ou no servidor, mostra unha mensaxe de erro ao usuario utilizando `alert()`.

2.2 Tarefa (tarefa.js)

O obxectivo desta apartado é implementar un script en JavaScript que cargue dinámicamente unha lista de tarefas ao facer clic na ligazón "Tarefas" do panel de usuario. O script debe realizar unha petición ao servidor para obter os datos das tarefas e mostralos nunha táboa dentro da páxina sen recargar.

Inicio Perfil Tarefa Sair			
tarefas			
ID	Produto	Cantidade	Prezo Data
1	Estudar para o exame de Matemáticas	Revisar os capítulos 3 e 4 do libro.	Alta 2025-04-15
2	Completar o proxecto de Historia	Preparar unha presentación sobre a Revolución Industrial.	Media 2025-04-20
3	Enviar o traballo de Inglés	Escribir un ensaio sobre o impacto da tecnoloxía.	Baixa 2025-04-25
4	Preparar a exposición de Ciencias	Facer un modelo do sistema solar.	Alta 2025-04-30
5	Revisar para o exame de Historia	Estudar os principais eventos da Segunda Guerra Mundial.	Media 2025-05-05
6	Fazer o traballo de Arte	Crear unha pintura inspirada na obra de Van Gogh.	Baixa 2025-05-10
7	Completar a tarefa de Educación Física	Practicar para a competición de atletismo.	Alta 2025-05-15
8	Enviar o traballo de Filosofía	Escribir un ensaio sobre a ética de Aristóteles.	Media 2025-05-20

Requisitos

1. Ligazón "Tarefas":

- O script debe detectar cando o usuario fai clic na ligazón que apunta a [tarefa.html](#).
- O comportamento predeterminado da ligazón (navegar a outra páxina) debe ser evitado.

2. Petición ao servidor:

- Realiza unha petición **GET** ao endpoint **api/tarefa.php**.
- Inclúe as cookies na petición utilizando **credentials: "include"** para xestionar a sesión do usuario.

3. Resposta do servidor:

- O servidor devolverá unha resposta en formato **JSON** cunha lista de tarefas. Cada tarefa terá o seguinte formato:

```
{
  "id_tarefa": 1,
  "titulo": "Estudar para o exame de Matemáticas",
  "descripcion": "Revisar os capítulos 3 e 4 do libro.",
  "prioridade": "Alta",
  "data_limite": "2025-04-15"
}
```

4. Mostrar as tarefas:

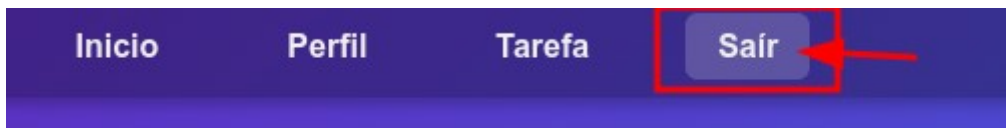
- O script debe crear unha táboa para mostrar as tarefas.
- Cada tarefa debe ser representada nunha fila da táboa con estas columnas:
 - **ID:** Identificador único da tarefa.
 - **Título:** Nome ou título da tarefa.
 - **Descrición:** Breve descrición da tarefa.
 - **Prioridade:** Nivel de prioridade (Alta, Media, Baixa).
 - **Data límite:** Data límite para completar a tarefa.

5. Xestión de erros:

Se ocorre un erro na petición ou no servidor, mostra unha mensaxe de erro ao usuario utilizando `alert()`.

2.3 Cerrar a sesión (logout.js)

O obxectivo desta tarefa é implementar un script en JavaScript que xestione a autenticación do usuario e permita finalizar a sesión (logout). O script debe verificar se o usuario está autenticado ao cargar a páxina e redirixilo á páxina de login se non o está. Ademais, debe implementar a funcionalidade de logout para finalizar a sesión e redirixir ao usuario á páxina de inicio de sesión.



Requisitos:

1. Verificar a autenticación do usuario:
 1. Ao cargar a páxina, realiza unha petición GET ao endpoint **api/check_session.php**.
 2. Se o usuario non está autenticado (logged_in: false), redirixe á páxina de login (`index.html`).
2. Implementar o logout:
 1. Engade un event listener ao botón ou ligazón de logout (co ID `logout`).
 2. Ao facer clic no botón de logout:
 1. Realiza unha petición **GET** ao endpoint **api/logout.php** para finalizar a sesión
 2. Redirixe ao usuario á páxina de login (`index.html`).
3. Mensaxes de erro: Se ocorre un erro na petición, podes engadir unha mensaxe de erro na consola para facilitar a depuración.
4. Obxectivo final, ao completar esta tarefa, o script debe:
 1. Verificar se o usuario está autenticado ao cargar a páxina e redirixilo á páxina de login se non o está.
 2. Permitir ao usuario finalizar a sesión ao facer clic no botón de logout e redirixilo á páxina de login.

Puntuación

Apartado	Puntuación
1.1 Login.js	10
2.1 Perfil (profile.js)	10
2.2 Tarefa (tarefa.js)	10
2.3 Cerrar a sesión (logout.js)	10