

Exercicio 80: Creación dunha app de resultados deportivos

Obxectivo: Proponse deseñar e desenvolver unha aplicación móbil híbrida para Android, empregando Capacitor JS, que amose a información do último partido xogado e a clasificación actual dun equipo deportivo.

Requisitos:

- Para o **deseño tes total liberdade**: podes deseñar un, podes usar un framework de CSS ou usar esta proposta de base: [Deseño en Figma](#).
- O usuario pode escoller libremente o equipo do que desexa amosar a información, tanto de baloncesto (ACB) como de fútbol (Primeira División española ou Premier League).
- A app debe consultar a [API pública de TheSportsDB](#) para obter:
 - Os detalles do último partido xogado polo equipo seleccionado (data, rival, marcador)..
- A información debe presentarse cun **deseño responsive** e atractivo.

Endpoints necesarios para facer o fetch (exemplo para Monbus Obradoiro, adaptables a calquera equipo):

```
// Buscar o ID do equipo (substitúe "Monbus Obradoiro" polo nome do equipo desexado)
https://www.thesportsdb.com/api/v1/json/3/searchteams.php?t=Monbus%20Obradoiro

// Obter detalles do último partido xogado (substitúe {ID_EQUIPO} polo ID obtido)
https://www.thesportsdb.com/api/v1/json/3/eventslast.php?id={ID_EQUIPO}

// Obter a clasificación actual da liga (ID da liga: 4408 para ACB, 4335 para LaLiga)
https://www.thesportsdb.com/api/v1/json/3/lookupstable.php?l={ID_LIGA}&s=2024-2025
```

Exemplo de fetch para resultados do último partido:

```
fetch('https://www.thesportsdb.com/api/v1/json/3/searchteams.php?t=Obradoiro%20CAB')
  .then(res => res.json())
  .then(data => {
    if (!data.teams) {
      document.getElementById('resultado').innerHTML = 'Equipo no encontrado';
      return;
    }
  })
```

```

    }

    const equipo = data.teams[0];

    const idEquipo = equipo.idTeam;

    obtenerUltimoPartido(idEquipo, equipo.strTeam);

  });

function obtenerUltimoPartido(idEquipo, nombreEquipo) {

  fetch(`https://www.thesportsdb.com/api/v1/json/3/eventslast.php?id=${idEquipo}`)

    .then(res => res.json())

    .then(data => {

      if (!data.results) {

        document.getElementById('resultado').innerHTML = 'Non hai partidos disponibles';

        return;

      }

      const partido = data.results[0];

      mostrarPartido(partido, nombreEquipo);

    });

}

function mostrarPartido(partido, nombreEquipo) {

  const contenedor = document.getElementById('resultado');

  const rival = (partido.strHomeTeam === nombreEquipo) ? partido.strAwayTeam : partido.strHomeTeam;

  const localia = (partido.strHomeTeam === nombreEquipo) ? "Na casa" : "Fóra";

  contenedor.innerHTML = `

    <h2>Último partido de ${nombreEquipo}</h2>

    <p><strong>Data:</strong> ${partido.dateEvent}</p>

    <p><strong>Rival:</strong> ${rival}</p>

    <p><strong>Marcador:</strong> ${partido.intHomeScore} - ${partido.intAwayScore}</p>

    <p><strong>Localía:</strong> ${localia}</p>

  `;

}

```

Notas:

- Recoméndase revisar o nome exacto do equipo na base de datos de [TheSportsDB](#) para garantir que a consulta devolva resultados.
- A app debe funcionar correctamente nun dispositivo Android e adaptarse a diferentes tamaños de pantalla.

EXTRA (opcional):

- A clasificación actual da liga correspondente

Entrega: Unha vez completado todo **subiremos a carpeta *Exercicio80*** ao espazo para entrega de exercicios que temos no OneDrive de Microsoft Teams.

Ejercicio 80: Creación de una app de resultados deportivos

Objetivo:

Se propone diseñar y desarrollar una aplicación móvil híbrida para Android, utilizando Capacitor JS, que muestre la información del último partido jugado y la clasificación actual de un equipo deportivo.

Requisitos:

- Para el diseño tienes total libertad: puedes crear uno propio, puedes usar un framework de CSS o utilizar esta propuesta como base: [Diseño en Figma](#).
- El usuario puede escoger libremente el equipo del que desea mostrar la información, tanto de baloncesto (ACB) como de fútbol (Primera División española o Premier League).
- La app debe consultar la [API pública de TheSportsDB](#) para obtener:
 - Los detalles del último partido jugado por el equipo seleccionado (fecha, rival, marcador).
- La información debe presentarse con un diseño responsive y atractivo.

Endpoints necesarios para hacer el fetch (ejemplo para Monbus Obradoiro, adaptables a cualquier equipo):

```
// Buscar el ID del equipo (sustituye "Monbus Obradoiro" por el nombre del equipo deseado)
```

```
https://www.thesportsdb.com/api/v1/json/3/searchteams.php?t=Monbus%20Obradoiro
```

```
// Obtener detalles del último partido jugado (sustituye {ID_EQUIPO} por el ID obtenido)
```

```
https://www.thesportsdb.com/api/v1/json/3/eventslast.php?id={ID_EQUIPO}
```

```
// Obtener la clasificación actual de la liga (ID de la liga: 4408 para ACB, 4335 para LaLiga)
```

```
https://www.thesportsdb.com/api/v1/json/3/lookuptable.php?l={ID_LIGA}&s=2024-2025
```

Ejemplo de fetch para resultados del último partido:

```
fetch('https://www.thesportsdb.com/api/v1/json/3/searchteams.php?t=Obradoiro%20CAB')
  .then(res => res.json())
  .then(data => {
    if (!data.teams) {
      document.getElementById('resultado').innerHTML = 'Equipo no encontrado';
      return;
    }
  })
```

```

    const equipo = data.teams[0];
    const idEquipo = equipo.idTeam;
    obtenerUltimoPartido(idEquipo, equipo.strTeam);
  });

function obtenerUltimoPartido(idEquipo, nombreEquipo) {
  fetch(`https://www.thesportsdb.com/api/v1/json/3/eventslast.php?id=${idEquipo}`)
    .then(res => res.json())
    .then(data => {
      if (!data.results) {
        document.getElementById('resultado').innerHTML = 'No hay partidos disponibles';
        return;
      }
      const partido = data.results[0];
      mostrarPartido(partido, nombreEquipo);
    });
}

function mostrarPartido(partido, nombreEquipo) {
  const contenedor = document.getElementById('resultado');
  const rival = (partido.strHomeTeam === nombreEquipo) ? partido.strAwayTeam : partido.strHomeTeam;
  const localia = (partido.strHomeTeam === nombreEquipo) ? "En casa" : "Fuera";
  contenedor.innerHTML = `
    <h2>Último partido de ${nombreEquipo}</h2>
    <p><strong>Fecha:</strong> ${partido.dateEvent}</p>
    <p><strong>Rival:</strong> ${rival}</p>
    <p><strong>Marcador:</strong> ${partido.intHomeScore} - ${partido.intAwayScore}</p>
    <p><strong>Localía:</strong> ${localia}</p>
  `;
}

```

Notas:

- Se recomienda revisar el nombre exacto del equipo en la base de datos de [TheSportsDB](#) para garantizar que la consulta devuelva resultados.
- La app debe funcionar correctamente en un dispositivo Android y adaptarse a diferentes tamaños de pantalla.

EXTRA (opcional):

- La clasificación actual de la liga correspondiente.

Entrega:

Una vez completado todo, subiremos la carpeta ***Ejercicio80*** al espacio para entrega de ejercicios que tenemos en el OneDrive de Microsoft Teams.