

TP 2 🏆 Cuatro meses de emociones y una PWA

Objetivo: Poder cumplir con el programa visto en clase, realizando una PWA.

Funcionalidad básica:

Como usuario, quiero poder entrar a una aplicación web que consuma un servicio REST. Esta aplicación debe poder ser instalada, y poder ofrecerme funcionar de manera "desconectada" (offline).

La aplicación debe estar "deployada" en Vercel (<https://vercel.com/dashboard>)

Por favor, lean el TP hasta el final, y cualquier inquietud enviar por email.

Fecha de Entrega: Clase del 01.07.2021

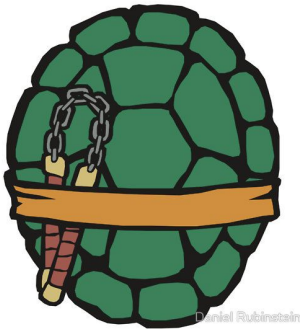
Enviar email a gonzalo.rubino@davinci.edu.ar con:

- Link al sitio deployado en Vercel
- Link al repo de Github
- El Readme del repositorio debe incluir Nombre Completo, Comision, y alguna instrucción para correr la APP si hubiese.
- Agregar por favor el camino elegido.

Los puntos de calificación son referencia, y de faltar alguna de los requerimientos enumerados será considerado un porcentaje del puntaje total.

Application Shell (

<https://developers.google.com/web/fundamentals/architecture/app-shell?hl=es>)



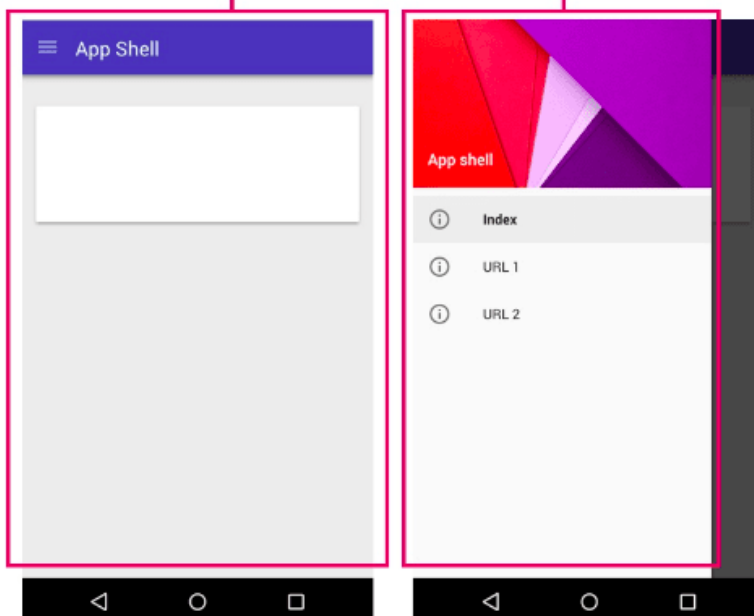
Básico: 4 puntos

La estructura base

La aplicación debe contener un “Shell” que pueda verse offline (gracias al Service Worker).

- Service Worker (1 punto)
Si la aplicación se encuentra offline (no tiene conexión), debo poder ver el homepage, y tener un mensaje de aviso de offline.
- Shell básico (navbar, que pueda usarse para navegar de un lado a otro) (1 punto)
Mínimo de dos URL => Homepage y una mas
- Deploy en Vercel con HTTPS (1 punto)
- Responsive (1 punto).
- La app debe tener TITLE => Nombre de Alumno, Cátedra PWA
- FAVICON que la identifique

application shell



Cached shell loads **instantly** on repeat visits.

content

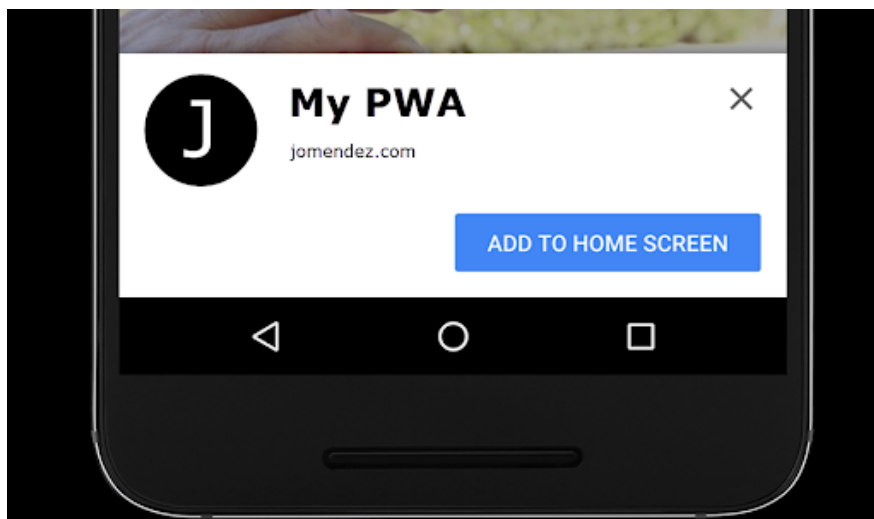


Dynamic content then populates the view



Quiero tener una casa: 2 puntos
La aplicación debe poder instalarse

Los usuarios pueden instalar la app desde Mobile o desde Desktop (Add to Home Screen) (2 puntos).



Camino al éxito: 4 puntos
Elige tu propia aventura.

Usando una de las siguientes API, la aplicación debe mostrar contenido en la vista principal (Content Area) y ofrecerle al usuario alguna funcionalidad.

Camino A (Futbol): <https://www.scorebat.com/video-api/>

Se hace un request a <https://www.scorebat.com/video-api/v1/>

Se van a devolver una serie de videos de partidos de futbol. Van a mostrar esas thumbnails.

El usuario puede elegir uno de ellos, y eso va a disparar que se muestre el video relacionado en el area principal.

Camino B (Musica): https://www.theaudiodb.com/api_guide.php

Input de búsqueda, para buscar un artista. El valor ingresado le pega a la API proporcionada arriba. El resultado se presenta en el content area.

Datos a mostrar: Biografía, una foto del grupo, genero musical, website, redes sociales.

Camino C (Anime / Manga): <https://jikan.moe/>

Input de búsqueda, para buscar un Anime. El valor ingresado le pega a la API proporcionada (ej: <https://api.jikan.moe/v3/search/anime?q=dragon%20ball>) .

El resultado se presenta en el content area.

Datos a mostrar: Imagen, titulo, sinopsis, score, link a myanimelist (clickable).

Camino D (Películas) Precisan API Key:

<http://www.omdbapi.com/>

o <https://www.themoviedb.org/documentation/api>

Input de búsqueda, para buscar una película. El valor ingresado le pega a la API proporcionada (ej: <http://www.omdbapi.com/?apikey=XXX&t=the+matrix&plot=full>) .

El resultado se presenta en el content area.

Datos a mostrar: Imagen, titulo, sinopsis, score, link al sitio (si hubiera).



En el estado offline, la aplicación tiene “algo” que entretiene o le es útil al usuario. Queda a criterio del alumno. SorprendanmeN.

Pero Profe, si saque 10, ¿para qué quiero el punto adicional?

Bueno, pero puede sumar al promedio general y que apruebes la cursada.

O puede ser un importante punto para tu desarrollo profesional como programador.

IMPORTANTE:

Algo que decimos en software: **“Make it work, make it better”**

Hay varias frases al respecto de lo mismo. Pero la idea es hacerlo funcionar y después mejorarlo a nivel código.

En la evaluación se revisará el aspecto funcional (que funcione correctamente) y también el lado técnico (como fue desarrollado, porque se tomó esas decisiones, si usan un framework porque fue... etc).

Hecho el correspondiente “disclaimer”: **A TRIUNFAR!**

Fecha de Entrega: Clase del 01.07.2021

Enviar email a gonzalo.rubino@davinci.edu.ar con:

- **Link al sitio deployado en Vercel**
- **Link al repo de Github**
- **El Readme del repositorio debe incluir Nombre Completo, Comision, y alguna instrucción para correr la APP si hubiese.**
- **Agregar por favor el camino elegido.**