**PROYECTO 3: Tablero de Datos ("Dashboard")**

**ÍNDICE**

* [1. Intro](https://github.com/U-Camp/BOOT-M2-SEM13-PROY3#1-intro)
* [2. Demo](https://github.com/U-Camp/BOOT-M2-SEM13-PROY3#2-demo)
* [3. ¿Qué construirás?](https://github.com/U-Camp/BOOT-M2-SEM13-PROY3#3-qu%C3%A9-construir%C3%A1s)
* [4. Objetivos de Aprendizaje](https://github.com/U-Camp/BOOT-M2-SEM13-PROY3#4-objetivos-de-aprendizaje)
* [5. Requisitos](https://github.com/U-Camp/BOOT-M2-SEM13-PROY3#5-requisitos-y-entregables)
* [6. Criterios de evaluación](https://github.com/U-Camp/BOOT-M2-SEM13-PROY3#6-criterios-de-evaluaci%C3%B3n)
* [7. Entregas](https://github.com/U-Camp/BOOT-M2-SEM13-PROY3#7-entregas)

**1. Intro**

Uno de los beneficios que profesionales en desarrollo pueden dar a sus equipos es la **comunicación de datos**.

El desarrollo de tableros es una constante en los proyectos de software.

Incluso, han aparecido "dashboards" fáciles de gestionar a través de servicios como Google Studio que te permiten integrar datos de manera efectiva.

Por ello, aprender a integrar datos en nuestras aplicaciones, ya sea para un usuario, un equipo o un "stakeholder", y comunicarlas correctamente, es un ejercicio sólido para practicar habilidades de programación.

En este proyecto, trabajaremos con una librería de gráficas muy popular, llamada Chart.js.

Adicionalmente, harás una conexión a través de un API a una base de datos controlada por un tercero. Una vez descargados los datos en nuestra aplicación podrás manipularlos para expresarlos como desees.

**2. Demo**

Puedes entrar a ver el demo en este vínculo: <https://cocky-wescoff-e5cf09.netlify.app/>

* En este ejemplo, observarás una conexión vía API a [fixer.io](https://fixer.io/), el cual es un servicio API que ofrece en tiempo real el valor de una moneda con respecto a otras. Por lo tanto, la gráfica es real, establecida al día de hoy.
* Observa que tenemos estructurada la aplicación bajo la influencia del ambiente de desarrollo Node.js.
* Se utilizó asincronía para la recepción de datos y luego, la invocación del gráfico.

Puedes revisar el proyecto en la carpeta demodentro de este repositorio.

**3. ¿Qué construirás?**

En este proyecto construirás una aplicación que consuma una API. Puede ser pública o privada.

Te recomendamos revisar este repositorio en GitHub para elegir un API que pueda ayudarte a desarrollar este proyecto: <https://github.com/public-apis/public-apis>

Estos son las características a considerar:

* Usar un ambiente de desarrollo con Node.js
* Utilizar [Chart.js](https://www.chartjs.org/) para el desarrollo de gráficas. En caso de quere utilizar una diferente, es posible.
* Realizar una conexión a una API externa para extraer datos. Opcionalmente, puedes utilizarla con una API KEY.

**4. Objetivos de aprendizaje**

* Crear un ambiente de desarrollo con Node.js
* Conocer el funcionamiento de las APIs
* Aplicar procesos síncronos y asíncronos en JavaScript

**5. Requisitos y entregables**

Como principal requisito, este deberá realizarse en parejas (por motivos específicos, podría trabajarse individualmente).

A continuación, usarás esta lista para saber los requisitos mínimos del proyecto:

**UI**

*  Aplicar HTML5.
*  Aplicar selectores en CSS.
*  Sección de gráficos (Chart.js utiliza canvas).

**JAVASCRIPT**

*  Uso de fetch o axios para el consumo de datos.
*  Modularización con import y export.
*  Uso de promesas o async-await.

**CONTROL DE VERSIONES**

*  Crear un repositorio en GitHub y Realizar mínimo 5 "commits" por persona en tu repositorio.
*  Crear un README.md dentro del repositorio. Incluir la descripción, instalación y uso del proyecto.

**DESPLIEGUE**

*  Crear una URL para compartir (puedes utilizar Netlify o GitHub Pages).
*  Colocar descripción dentro del repositorio e incluir URL.

**ENTREGA A TIEMPO**

*  Entregar tu proyecto en el tiempo estipulado.

**6. Criterios de evaluación**

A partir de los requisitos, tu % estará definido de esta forma:

| **ÁREA** | **% DEL TOTAL** |
| --- | --- |
| UI | 24% |
| JAVASCRIPT | 45% |
| CONTROL DE VERSIONES | 10% |
| DESPLIEGUE | 6% |
| ENTREGA A TIEMPO | 15% |

**7. Entregas**

* **Definición de fechas.** Habla con tus coaches sobre el periodo de entregas del proyecto.
* **Cumple con los requisitos y entregables.** En caso de que no logres cumplir con todos los requisitos, el % que obtendrás responderá a los criterios de evoluación. Es importante que entregues tu proyecto a tiempo.