



### Tecnologías a utilizar:

- Framework: Laravel 10
- PHP 8.1
- JAVASCRIPT
- CSS
- MYSQL

### Desarrolle el siguiente sistema:

Se requiere el desarrollo de un sistema que permita utilizar la api de [google speed page insights](#) para mostrar las métricas principales de las categorías seleccionadas. Los requerimientos son los siguientes:

1- Crear 3 modelos:

- **MetricHistoryRun:** id, url, accesibility\_metric, pwa\_metric, performance\_metric, seo\_metric, best\_practices\_metric (con sus respectivos created and updated dates)
- **Category:** id, name (crear un seeder con los valores iniciales: ACCESSIBILITY, BEST\_PRACTICES, PERFORMANCE, PWA, SEO)
- **Strategy:** id, name (crear un seeder con los valores iniciales: DESKTOP, MOBILE)

Los modelos **MetricHistoryRun** y **Strategy** tienen que estar relacionados, cada run se ejecuta solo con una estrategia.

2-Crear una vista (con blade) en donde se muestren los siguientes campos:

- **input text:** Donde se podrá ingresar una URL para obtener las métricas
- **grupo de checkboxes:** Donde se mostrarán todas las categorías (1 o múltiples)
- **select:** donde se elegirá una única estrategia.
- **submit button:** Para ejecutar el proceso que usa la api y obtiene las métricas.

3-Crear un método en laravel (que pueda ser llamado desde la vista) que realice la llamada al servicio de Google utilizando Guzzle, creando un cliente específico para la llamada (no usar el Facade Http de Laravel). Desde la vista realizar la consulta vía AJAX con JAVASCRIPT (puro o jQuery, como prefieras) a este método creado previamente.

El método de Laravel debe utilizar la siguiente url para obtener los datos.

GET:

<https://www.googleapis.com/pagespeedonline/v5/runPagespeed?url=URL&key=AlzaSyDCrPAzhzWxZbJxPYIEURODTvBFVVRNHbY&category=CATEGORY1&category=CATEGORY..N&strategy=STRATEGY>

### **Ejemplo:**

<https://www.googleapis.com/pagespeedonline/v5/runPagespeed?url=https://broobe.com&key=AlzaSyDCrPAzhzWxZbJxPYIEURODTvBFVVRNHbY&category=PERFORMANCE&category=SEO&c>



ategy=BEST\_PRACTICES&category=ACCESSIBILITY&strategy=MOBILE

4-Luego de realizar la consulta vía AJAX, debajo de los campos detallados en el punto anterior se deben mostrar las métricas obtenidas con los valores devueltos por la api de google, mostrando "nombre de métrica" y "valor obtenido" (y lo que se requiera a nivel grafico). Del JSON devuelto por la consulta, Los resultados se obtienen del score de cada categoría, como se muestra en la imagen:

```
- lighthouseResult: {
  requestedUrl: "https://broobe.com/",
  finalUrl: "https://www.broobe.com/",
  mainDocumentUrl: "https://www.broobe.com/",
  finalDisplayedUrl: "https://www.broobe.com/",
  lighthouseVersion: "11.0.0",
  userAgent: "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) HeadlessChrome/116.0.5:
  fetchTime: "2023-08-24T11:48:57.375Z",
+ environment: { ... },
+ runWarnings: [ ... ],
+ configSettings: { ... },
+ audits: { ... },
- categories: {
  - performance: {
    id: "performance",
    title: "Performance",
    score: 0.45,
    + auditRefs: [ ... ]
  },
  - accessibility: {
    id: "accessibility",
    title: "Accessibility",
    description: "These checks highlight opportunities to [improve the accessibility of your web app]",
    score: 0.82,
    manualDescription: "These items address areas which an automated testing tool cannot cover. Learn
    + auditRefs: [ ... ]
  },
  - best-practices: {
    id: "best-practices",
    title: "Best Practices",
    score: 0.95,
    + auditRefs: [ ... ]
  },
  - seo: {
    id: "seo",
    title: "SEO",
    description: "These checks ensure that your page is following basic search engine optimization ad
    score: 0.93,
    manualDescription: "Run these additional validators on your site to check additional SEO best pra
    + auditRefs: [ ... ]
  }
},
```

5-Adicionalmente a los resultados, también se debe mostrar un botón que diga "Save Metric Run". Este botón, via ajax guardará los datos del modelo **MetricHistoryRun**, relacionados a **Strategy**.

6-Hacer una vista donde se muestra un listado de todo el historial en una tabla con las siguientes columnas:

URL	ACCESIBILITY	PWA	SEO	PERFORMANCE	BEST PRACTICES	STRATEGY	DATETIME
https://broobe.com	0.82		0.93	0.45	0.95	MOBILE	01-01-2023 09:00:00



Mockup de la pantalla principal luego de correr el servicio:

**IMPORTANTE:** Es solo a modo ilustrativo, en el challenge se evaluará el diseño aplicado para poder definir el nivel en CSS, manejo de efectos, manipulación del DOM, usabilidad, interacción con el usuario, etc. Se puede (y se sugiere) cambiar la estructura de la pantalla:

## Broobe Challenge

Run Metric

Metric History

URL

CATEGORIES

☐ PWA

☒ SEO

☒ PERFORMANCE

☒ BEST-PRACTICES

☒ ACCESIBILITY

STRATEGY

Mobile

GET METRICS

SEO0.93

PERFORMANCE0.45

BEST PRACTICES0.95

ACCESIBILITY0.82

SAVE METRIC RUN

**Importante:** Este es solo un mockup para reforzar la descripción del ejercicio. El diseño que se utilice es parte de lo que se evaluará en cuanto al uso de CSS.

Formato de entrega:

- Se debe entregar un link de github para poder clonar el proyecto y probarlo (incluyendo el .env)
- El proyecto debe tener un archivo readme indicando lo que se considere necesario para la instalación.

Puntos que tendremos en cuenta para evaluar el challenge:

- Prolijidad del código respetando el patrón MVC que promueve Laravel
- Separación de lógica de negocio de los controladores
- Validaciones back y front end
- Utilización de funcionalidades que brinda el framework para mejorar el producto final (ejemplo: Validators, Translations)



- Optimización del código
- Arquitectura (estructura, modularización, reutilización)
- Calidad del código en cuanto a buenas prácticas de programación
- Creatividad en la estructura y el diseño
- Decisiones sobre la interacción de la aplicación con el usuario (ejemplo: mostrar o no loadings, mensajes de feedback, etc.)
- Se deben utilizar plugins de JS para mejorar la usabilidad e interacción con el usuario (ejemplo; Datatables, select2, **mostrar los valores de las métricas utilizando un plugin de gráficos en vez de solo un número**, etc).

#### **Distribución del peso en la evaluación**

- Uso de laravel de forma óptima y con buenas prácticas, mostrando skills de backend: **45%**
- Creatividad en el front end en cuanto a UX/UI: **40%**
- Detalles extras a lo solicitado para validar seniority: **15%**