

# Informe - Cypress Challenge

## Descripción general del proyecto:

El objetivo del proyecto Cypress Challenge es crear una aplicación Cypress para realizar pruebas automatizadas en la página web de CallendarApp. El desafío consiste en recorrer la página y analizar su comportamiento, específicamente marcando el "Día de la Marmota" el 2 de febrero.

## Requisitos:

- Crear un test case que abra la página.
- Recorrer los meses hasta llegar a febrero (no importa en qué mes empiece).
- Verificar si el 2 de febrero ya existe el "Día de la Marmota" y crearlo si no existe.
- Crear un test case que borre el registro del "Día de la Marmota".

## Arquitectura y estructura:

La arquitectura del proyecto se basa en Cypress 12, utilizando Node.js 16 como plataforma.

## Instalación y configuración:

Para configurar el proyecto, se deben seguir los siguientes pasos:

- Clonar el repositorio del proyecto usando el comando **git clone**.
- Instalar todas las dependencias necesarias ejecutando **npm install**.
- Iniciar el proyecto Cypress en modo de desarrollo con el comando **npm run cypress:open**.

## Guía de uso:

- Una vez que el proyecto está configurado, se puede ejecutar **npm run cypress:open** para abrir el panel de Cypress.
- En el panel de Cypress, se puede seleccionar el archivo de prueba adecuado para ejecutar los casos de prueba.

- Por línea de comando se puede ejecutar el comando **npm run cypress:runcalendario** para ejecutar todas las pruebas del archivo `Calendario.cy.js`
- Por defecto, está configurado que después de ejecutar **npm run cypress:runcalendario** o **npm run cypress:run**, se genere un archivo de reporte que contenga los resultados de las pruebas. Dichas pruebas están en la carpeta **mochawesome-report** en formato HTML.
- Cuando se ejecuta **npm run cypress:runcalendario** o **npm run cypress:run** se generará un video mostrando los test ejecutados y los resultados de los mismos. El video generado se guardará en la carpeta **videos**.

## Ejemplos de código:

A continuación, se presentan ejemplos de código para los casos de prueba mencionados:

- Abrir la página:

```
before('Abrir la pagina como iniciador', () =>{
  cy.visit('https://calendar-challenge-six.vercel.app/')
})
```

- Recorrer los meses hasta febrero:

```
it('@commandexpres]Llegar hasta febrero', () => {
  //Esta funcion recorre todos los meses hasta llegar a febrero estes donde
  estes
  cy.ValidarMes()
})
```

- Verificar y crear el "Día de la Marmota" el 2 de febrero:

```
it('@commandexpres]Crear Nota Dia de la Marmota', () => {
  //Esta funcion agrega la nota "El Dia de la Marmota" el dia 2 de febrero
  cy.ValidarDiaDeLaMarmota()
})
```

- Borrar el registro del "Día de la Marmota":

```
it('@borrardia] Borrar Nota Dia de la Marmota', () => {  
  //Esta funcion borra la nota previamente creada en el dia de 2 de febrero  
  cy.BorrarDiaDeLaMarmota()  
})
```

El resto del código y los procedimientos están en el archivo **commands.js**

## Pruebas:

La estrategia de pruebas sugerida incluye los siguientes casos de prueba:

- Prueba de apertura de la página.
- Prueba de recorrido de meses hasta febrero.
- Prueba de verificación y creación del "Día de la Marmota".
- Prueba de borrado del registro del "Día de la Marmota".

## Dependencias:

El proyecto utiliza las siguientes dependencias principales:

- Node.js 16
- Cypress 12

## Solución de problemas y preguntas frecuentes:

- Para este proyecto, tome la decisión de que las pruebas no sean aisladas, agregando como opción en el archivo **cypress.config.js** `testIsolation: false`, ya que era necesaria la información de estado entre prueba y prueba para la realización eficiente de las mismas.
- Como Cypress no permite ciclos iterativos como `do-while`, para recorrer todos los meses en la página, se optó por aplicar `recurrencia` en donde se verifica el mes, y si no es el mes de "febrero" se vuelve un mes para atrás.
- Para verificar la existencia de la nota dentro del 2 de febrero que diga "Dia de la marmota" se optó una solución contando las etiquetas `<div>` dentro del mismo día. Se realizó de esta manera porque en Cypress no se puede obtener el negativo de una aserción, y el error de ejecución no se puede controlar de manera manual.