

Estimados de complejidad

Servir sitio desde S3/CDN con Edge Cache

Esto es muy sencillo, simplemente en el momento de realizar la compilación, al terminar, copiar todo el sitio al bucket de donde se sirva la web.

- Asigno una complejidad de 10 o 20

Flexibilización de estilos basada en CSS frameworks extensibles

Agregar extensibilidad basada en frameworks extensibles y poco opinionadas como stylus + bulma no debería representar una dificultad muy grande. Creo que la mayor dificultad aquí es poder conectar todo de manera adecuada, hacer los diseños en sí y escribir las minilibrerías requeridas para lo que se quiera extender.

- Por lo tanto esto tiene una complejidad relativamente baja, quizás un 30 o 40.

Sistema de CI/CD y compilación automática de contenidos

Agregar compilación automática del sitio cuando se realiza el cambio de un contenido es algo simple y fácil de hacer con un monitor de cambios y un webhook que dispare el proceso, es algo que ya hacemos actualmente con el sitio a reemplazar.

Agregar un sistema de CI/CD general que pueda compilar *todo* es algo más complicado, pero usando algo como Drone, CircleCI o Jenkins no debería representar demasiada dificultad. Creo que el mayor desafío aquí es poder coordinar de manera adecuada cada etapa de compilación en los contenedores, y que el toolchain pueda ser algo complejo en función de cómo sean las utilidades y el backend.

- Por lo tanto esto tiene una complejidad media, quizás un 50 o 60.

Integración con Git/Mercurial/SVN

La integración en sí con un sistema de control de versiones es algo simple y directo, la dificultad aquí podría residir en cómo se relaciona el control de versiones con la forma en la que se almacenen los datos; si los datos se almacenan en el backend en una base de datos, habría que usar un sistema especial para bases de datos o de lo contrario escribir una capa de traducción que lea los datos, los traduzca a archivos rastreables por el sistema, y de la misma manera si se producen cambios en el repositorio, traducir al revés e incluir los datos en la database. Dependiendo de la complejidad de las estructuras de datos involucradas en la creación de los contenidos es to puede ser algo muy sencillo o algo muy complejo.

- Asigno una complejidad media-alta, 60 o 70, pero requiere que sea reevaluada más adelante cuando haya mejor información al respecto de cómo trabajar con esto.

Health endpoint de grafana

Esto es bastante straightforward, lo único que puede ser algo engorroso es recolectar los logs/datos con Prometheus, InfluxDB and the likes, y que sean consumidos por

grafana. En sí depende de qué tan profundo se quiera llegar, pero no debería hacer falta mucha profundidad si es solamente para estar al tanto del estatus de la página (aunque quizás si se agregan analíticas pueda ser algo más complejo).

- Asigno una complejidad baja, 30 o 40.

Integraciones basadas en webhooks para notificar sucesos, ediciones, etc

Esto debería ser relativamente sencillo, hay que monitorear cambios en la base de datos o en el endpoint de GraphQL y disparar el webhook. Los elementos a integrar, por ejemplo Mattermost, soportan webhooks de manera nativa, por lo que lo único que habría que hacer es código pegamento para relacionar las cosas entre sí.

- Asigno una complejidad baja, 30 o 40

Diseño de la estructura de datos base

Si bien no es demasiado difícil generar una estructura de datos viable para lo que esencialmente es un blog/portfolio de la organización, es importante acertar de entrada con esto, y creo que hace falta tomarse el tiempo para leer y procesar las cosas antes de hacer el diseño y pasarlo a esquemas de GraphQL.

- Asigno una complejidad media, 50 o 60

Sistema de visualización de datos basado en D3/Chart.js

Esto puede llegar a ser algo muy complicado en función de la complejidad propia de los datos a mostrar, el nivel de interactividad y el nivel de automatización requerida para mostrarlos. Creo que se pueden hacer plantillas para que muestren distintos tipos de gráficos con distintos niveles de interactividad, y agregarlas como campos personalizados en Strapi. Como GraphQL es muy flexible y poderoso, no habría problema en incorporar los esquemas para representar los gráficos de forma adecuada, y que React los consuma y transfiera a D3/Chart.js.

Por lo tanto si bien es claro cómo realizar este proceso, la complejidad puede ser elevada dependiendo de la profundidad con la que se quieran representar las cosas.

- Asigno un nivel alto variable de complejidad, 60 a 90

Roles y usuarios en el editor

Esto es soportado nativamente por Strapi así que debería ser muy simple.

- Asigno una complejidad muy baja, 10

Precompilación a SVG de fórmulas y LaTeX snippets por MathJax

Esto puede ser relativamente complicado dependiendo de cómo se aproxime el sistema que genere el SVG a mostrar. En el momento de la compilación del sitio se llamaría un microservicio que tome la data, la procese y coloque el SVG en su lugar. Creo que la dificultad aquí es poder manejar adecuadamente la compilación de LaTeX, MathJax y los quirks que puedan tener.

- Asigno una complejidad media, 50 o 60

Sistema de visualización de raw data; con algun framework para mostrar tablas y cosas así

Esto probablemente no sea demasiado complejo, ya que simplemente es mostrar tablas y hay librerías que tienen todo eso cocinado. La cosa sería integrarlas adecuadamente y agregar las opciones de visualización que se deseen. Estas opciones de visualización deberían ser algo muy simple; columna vs fila, fila vs fila, columna vs columna, sumas, diferencias, esas cosas.

- Asigno una complejidad media dependiendo de cómo se agregue la visualización de los datos crudos, 50 a 70

Minificación del frontend para minimizar el tiempo de carga/render de página

Esto puede ser todo un agujero de conejo de formas de optimizar y reducir el tamaño de todo. Creo que haciendo estático todo lo que se pueda, sirviendo desde un bucket, preprocesando lo que se pueda preprocesar, evitando tener demasiado JS en el cliente se puede llegar bastante lejos.

- Asigno una complejidad media, 40 o 50