Recomendaciones

24 de Abril 2024

ACID LABS

Recomendaciones sobre organización de proyecto

Utilizar versionamiento de releases, generar changelog

Utilizar versionamiento semantico (https://semver.org/) para notar cada release que pase a un ambiente productivo

- Asociado a cada release, generar un changelog de funcionalidades nuevas que hayan sido agregadas, mejoras, y otros cambios asociados.

Mejor control de recursos Javascript

Actualmente el Javascript, está fuertemente ligado a las vistas de cada uno de los ViewComponents. Esto se puede tornar difícil de mantener a medida que el código se torna complejo.

Recomendamos implementar las sugerencias propuestas por ViewComponent (https://viewcomponent.org/guide/javascript_and_css.html), encapsular cada asset JS/CSS en un archivo aislado. Y quizás considerar Stimulus para aquellos ViewComponent que tengan una alta interactividad con usuario.

Versionamiento de APIs

A modo de velar por la estabilidad de los clientes que consumen APIs, recomendamos utilizar un modelo de versionamiento de APIs:

- Encasillar el comportamiento de un API en un número de versión (ej: v2). Documentar las entradas y salidas esperadas por este servicio (se puede utilizar OpenAPI, o algo cómo una Postman collection).
- En caso de necesitar realizar cambios de comportamiento, estos deben ser retrocompatibles con los parámetros de entrada, y las salidas esperadas no deben presentar cambios.
    - Si se rompen estas condiciones, el cambio deberá ser comprendido en una nueva versión del API (ej: v3), notificando a los clientes sobre su modo de consumo y manteniendo ambas versiones del API en funcionamiento.

Considerar un control de infraestructura mediante código

Cuando la aplicación se despliegue en un ambiente productivo, consideramos buena práctica declarar la infraestructura, aprovisionamiento y servicios necesarios para montar el ambiente a modo de código.

De esta forma, la creación de ambientes queda automatizada y replicable. Se lleva un control de cambios en caso de existir nuevos requerimientos de infraestructura.

Algunas herramientas relacionadas:

- Terraform
- Ansible
- Puppet
- AWS CloudFormation

Velar por la calidad del código

Con los lineamientos de calidad de código ya establecidos (Rubocop, Rspec, Brakeman, Bundle Audit, etc) hemos podido mantener en control el estilo y comportamiento correcto del código, así mismo protegernos de vulnerabilidades de seguridad potenciales.

Recomendamos mantener la revisión estricta de estas herramientas (mediante pipeline, si es posible), y así mismo, continuar la cobertura de documentación y testing asociada a cada funcionalidad nueva a implementar.

Recomendaciones de mejora en funcionalidades

Sobre información del sitio de cara a usuario Solicitante

Informar a usuarios mediante el sitio, cómo este debe ser utilizado

- Aprovechar espacios en el homepage, para describir propósito y modo de uso de la plataforma.
- Ser conciso en el proceso de checkout y post-compra, sobre el estado en el cual está su orden
    - Agregar más comunicación por correo electrónico en caso de ser necesario.

Potenciar el buscador y la experiencia de navegación de productos

Permitir al buscador del sitio reconocer términos de búsqueda con errores ortográficos

- Ejemplo, poder buscar “ivuprofeno” y encontrar productos que calcen con “Ibuprofeno”.

Utilizar de mejor forma las opciones de filtrado del buscador

- Pensar en qué opciones de filtrado entregan valor a la hora de buscar productos
- Disponer opciones de ordenado

Mejorar el formato de la grilla de productos

- Mostrar más información del producto (ZGEN, ZCEN?)
- Permitir saber de antemano si el producto tiene stock o no

Agregar un arbol de categorias en el sitio

- Organizar los productos por categorías y subcategorías
- Generar un árbol de navegación que permite “filtrar” rápidamente en base a estas subcategorías.