

Technical Challenge Spot2 (IC FullStack)

En Spot2, una prop-tech pionera en el sector del real estate comercial, reconocemos que la innovación constante es el pilar de nuestro éxito. Esta dedicación a la vanguardia tecnológica nos exige no solo diseñar sistemas que escalen eficientemente y cumplan con los más altos estándares de calidad de código, sino también implementar soluciones que reflejen nuestra excelencia en la ejecución y nuestras sólidas prácticas de programación. Estamos buscando individuos apasionados por la tecnología que compartan nuestra dedicación a estos principios, para que juntos podamos continuar liderando la innovación en el mercado del real estate comercial.

Reto

Diseñar e implementar un URL-shortener haciendo uso de Laravel. Como máximo deben ser 8 caracteres fáciles de leer para el código de acortamiento.

Tiempo de desarrollo: 5 horas aprox.

Tiempo límite de entrega: 1 semana

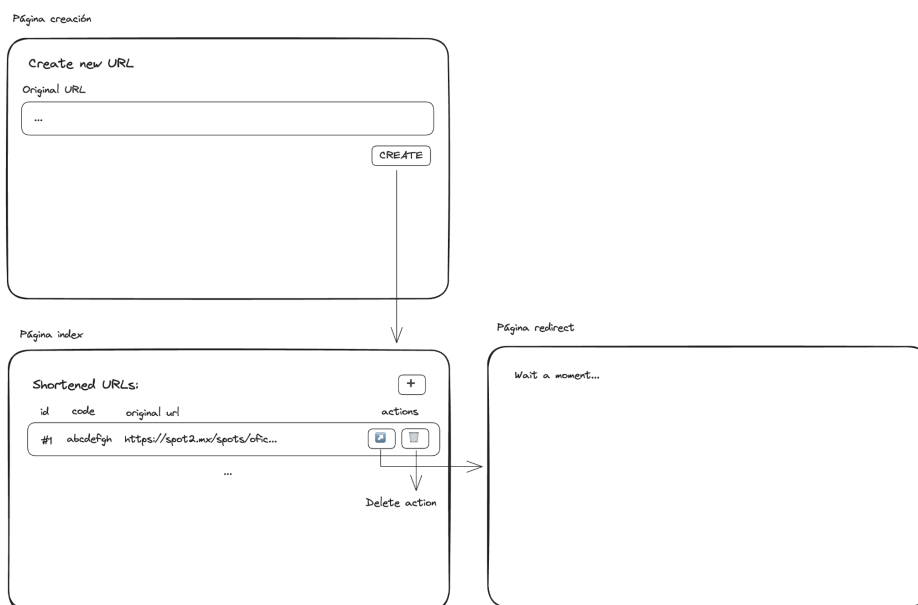
Requisitos

API RESTful:

- Documentación completa de la API usando OpenAPI (Swagger) o similar, incluyendo todos los endpoints, parámetros, y posibles respuestas.
- Cobertura de pruebas unitarias y de integración para el backend, asegurando la calidad y la fiabilidad del código. La metodología de Spot2 es TDD, así que sería bueno explicar en tu entrevista cómo fue tu proceso.

UI:

El flujo de un usuario puede ser como el que se ilustra a continuación: página de creación, index y redirect.



Despliegue:

- La aplicación se tiene que probar en un ambiente productivo.

Documentación:

- Proveer una guía de inicio rápido o un README detallado en el repositorio de GitHub, explicando cómo instalar, configurar y ejecutar la aplicación localmente y en la nube.
- Documentar cualquier decisión de diseño importante y justificar cómo estas decisiones apoyan los requisitos del negocio y los objetivos técnicos.

Consejos

- Este proyecto puede ser realizado utilizando el *free tier*¹ de AWS.
- Menos es más.
- Pensar fuera de la caja.
- Las migraciones y los seeders son herramientas muy útiles del proceso de desarrollo.
- ¿Por qué deberíamos adoptar tu solución o qué mejorarías?
- ¿Este sería un código que pondrías en producción?

Evaluación funcional

- Evaluación de la limpieza, mantenibilidad, y estructuración del código. Uso de patrones de diseño donde sea apropiado.
- La solución se va a probar acortando un URL. El URL acortado debe tener menos caracteres que el URL original.
 - Para efectos de la prueba, no se van a contar los caracteres del nombre del dominio (esto incluye dirección IP en caso de no tener nombre de dominio).
- Al acceder al URL acortado debe redireccionar a la URL original; ya sea porque regresa un HTTP 301 / 302 o porque regresa un JSON con el URL original.
- Se te van a realizar preguntas sobre cómo tu arquitectura podría escalar, y qué implicaciones podría tener dicha selección de arquitectura.
 - Estrategias para el manejo de configuraciones entre diferentes ambientes (desarrollo, pruebas, producción).
- Así mismo, también se te van a realizar preguntas sobre cómo se podrían desplegar nuevas versiones de la aplicación de forma automatizada (mediante un proceso de CI/CD).
- Innovación y uso creativo de tecnologías para resolver problemas de manera eficiente.

¿Dudas? Envíame correo a dante@spot2.mx

RFC sobre URIs <https://www.rfc-editor.org/rfc/rfc1738>

¹ <https://aws.amazon.com/free/>