

roseta de prioridades

creado 20190215T0929 guardado 20220214T1047 impreso 20220214T1047

En todos los proyectos, siempre hay una serie de prioridades. Estas prioridades conviene expresarlas con el verbo ‘aumentar...’ o con el verbo ‘reducir...’ delante de la frase que las define. Para ayudar a pensar mejor acerca de la relación entre prioridades, ya que:

- Algunas se alinean positivamente (cuando más nos acercamos a una de ellas, más nos acercamos también a la otra)
- Algunas se alinean negativamente (cuanto más nos acercamos a una de ellas, más nos alejamos de la otra)

Esto nos permite acabar representandolas en un mapa con forma de roseta. Donde se ve claramente que al potenciar las prioridades de una parte de la circunferencia, estamos relajando las prioridades en la parte opuesta:



En todo proyecto, es necesario tomar decisiones; y toda decisión implica adoptar ciertos compromisos. Esta roseta ayuda a ver claramente las implicaciones de dar más peso a unas prioridades o a otras.

La roseta pretende ser una herramienta para ayudar a pensar sobre esas implicaciones. Mostrando claramente cuales son nuestras asunciones y cual es nuestro posicionamiento respecto a la jerarquía de prioridades.

Forma de leerla:

- Prioridades en extremos opuestos de la circunferencia, se alinean de forma negativa entre ellas.
- Las prioridades más cercanas a la *marca de posicionamiento* ★ son las que se intentarán potenciar. (obviamente, a expensas de aquellas más lejanas).
- La marca de “*en caso de duda, es preferible resolverla relajando esta prioridad*” ☁ se puede utilizar para afinar el posicionamiento.

nota: Es obvio, pero nunca está de más aclararlo. Tener una jerarquía de prioridades no implica abandonar ninguna de ellas. Todas han de estar presentes en todas las decisiones y se ha de tratar de cumplir siempre con todas ellas. Pero, como a veces no es posible, es conveniente tener claros los criterios a aplicar en aquellas decisiones donde ciertas prioridades se alinean negativamente con otras y es necesario relajar algunas en favor de otras.

nota: Cuanto más cerca del centro se sitúe la marca de posicionamiento, más complicada será la toma de decisiones en el día a día de ese proyecto (más prioridades hay que volver a sopesar en cada decisión). Y viceversa, cuanto más alejada del centro esté la marca, en cualquier dirección, más se simplifica la toma de decisiones.

Un ejemplo:

Si se ha de decidir si ciertos datos conviene almacenarlos de forma centralizada en un servidor, o distribuidos entre varios servidores, o guardarlos localmente en cada dispositivo de usuario.

En estos dos escenarios, los posicionamientos potenciarían el almacenamiento en servidor:



Además, en este segundo, sería preferible la centralización.

En estos dos escenarios, el posicionamiento potenciaría el almacenamiento local:



nota: Es obvio, pero nunca está de más aclararlo. Si la ubicación del almacenamiento de esos datos es un requisito fijado por el cliente, no hay duda ni decisión a tomar. Tampoco la hay en el caso de que, por ejemplo, los datos a almacenar varíen cada pocos minutos o cada pocas horas y sea un requisito que el usuario trabaje siempre con datos actualizados. En el ejemplo se ha contemplado el caso de no tener ni una petición expresa ni una condición determinante para decidir dónde almacenar los datos.

rationale:

La idea tras esta herramienta es la de obligarnos a pensar en las dependencias entre prioridades y “mojarnos” en establecer una jerarquía de esas prioridades.

Fijar prioridades en cualquier proyecto es muy fácil. Normalmente suelen estar bastante claras. Por ejemplo en proyectos software siempre vamos a desear construir programas sencillos de usar, fáciles de mantener, con el máximo uso de elementos estándar, robustos, con menos costes, seguros, lo más parecidos posible a lo que ya sabemos usar,...

Pero todos sabemos que cumplir con todo eso a la vez no siempre es posible y que en algunos puntos vamos a tener que hacer concesiones. La idea de la roseta de prioridades es servir de instrumento para pensar acerca de las prioridades y llegar a plasmar criterios claros a la hora de realizar esas concesiones.

Por ejemplo, es muy fácil escribir las prioridades en cualquier parte de la circunferencia y situar la ‘marca de posicionamiento’ en el centro:



En la práctica, esto viene a significar: “en cada momento adoptaremos la decisión que nos parezca más conveniente en ese momento”.

Pero, por ejemplo, habiendo acordado un posicionamiento como este:



Estamos reconociendo que, por ejemplo, cuanto más potenciemos la seguridad de acceso, más coste económico supondrá. O que, cuando más sencillo de usar sea el programa para sus usuarios, más complejidad interna tendrá para sus programadores.

Y estamos dejando claro que, por ejemplo, preferimos la facilidad de uso, cambiar la vida lo menos posible a los usuarios, realizar trajes a medida para cada uno,...; frente a reducir costes, potenciar la facilidad de mantenimiento del software, utilizar paquetes estándares tal cual,...

Ojo: con estas preferencias no decimos que vayamos a gastar alocadamente, ni que vamos a despreocuparnos completamente de las buenas prácticas de programación que facilitan el mantenimiento del software, ni que vamos a desarrollar todo a medida partiendo de cero,...

Estamos diciendo simplemente que, por ejemplo, si para cumplimentar un requisito concreto hemos tomar una decisión entre:

- ‘formar a los usuarios en el uso del paquete estándar escogido e intentar modificar los procedimientos de trabajo para que se asemejen lo más posible a los de ese paquete estándar’

y

- ‘hacer las modificaciones necesarias en el paquete estándar para que encaje lo mejor posible con los procedimientos de trabajo actuales y lo que los usuarios ya conocen’

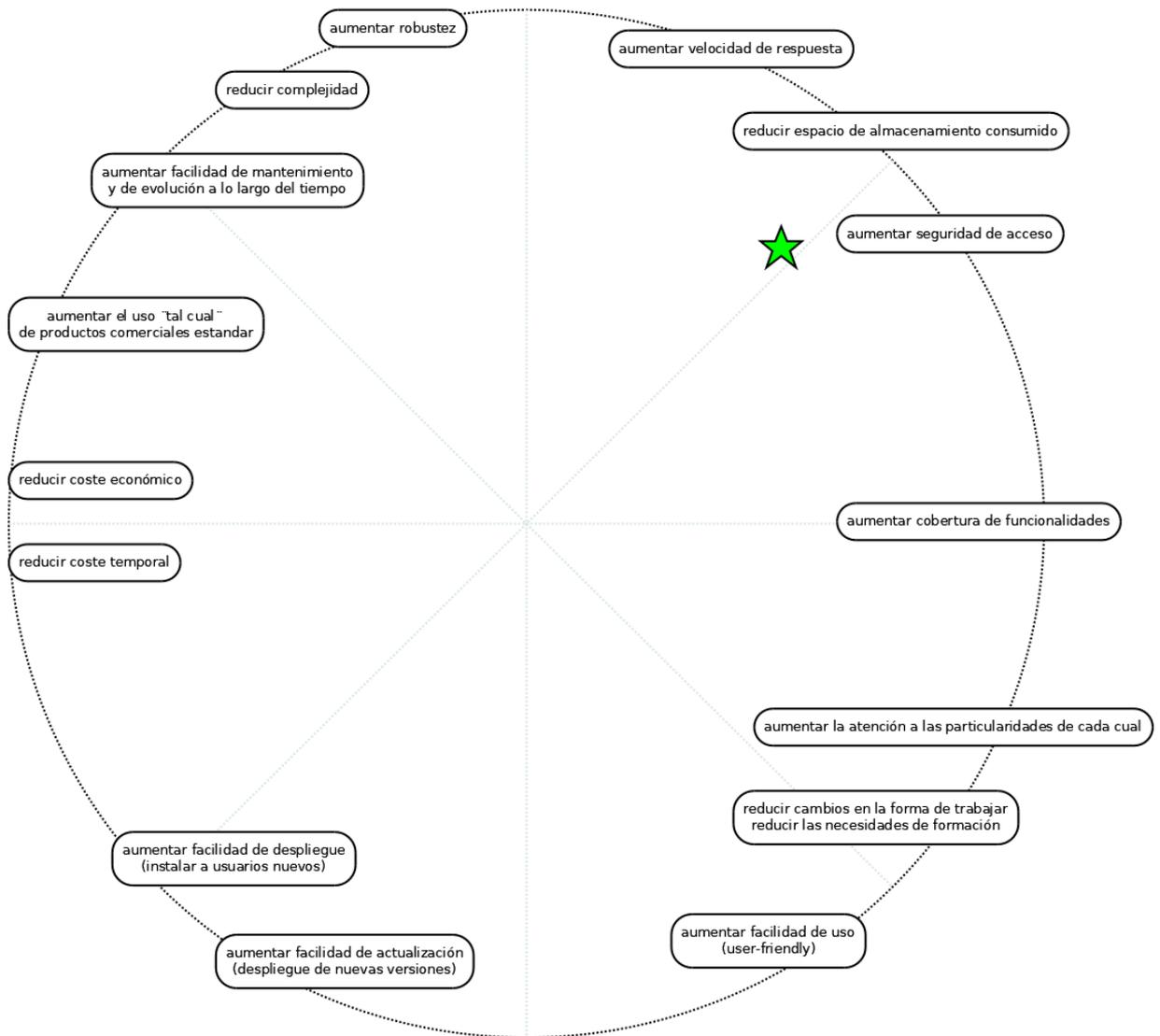
Vamos a priorizar siempre esta segunda opción.

Sin embargo, habiendo acordado un posicionamiento como este:



Nos encontraríamos en un escenario totalmente contrario, y tenderíamos siempre a priorizar la primera opción de entre las dos anteriormente citadas.

Y, por ejemplo, con un posicionamiento como este:



La decisión entre ambas opciones sería a favor de la primera si vemos que el paquete estándar, en esa funcionalidad, es más seguro que cualquier otra cosa que pudiéramos hacer nosotros; o a favor de la segunda si vemos que podemos hacer una modificación más segura que el funcionamiento estándar.

Es más, con este posicionamiento, hasta la misma selección de paquete estándar hubiera sido distinta. Habíamos buscado con criterios distintos, para escoger el paquete estándar más seguro que hubiéramos podido encontrar en el mercado.