**Introducción**

Con el presente documento se pretende definir una línea de desarrollo viable que nos permitiría alcanzar varias metas sucesivas:

1. Levantar un portal corporativo
2. Generar una maqueta/prototipo de la aplicación para facilitar la venta.
3. Completar una primera versión funcional del producto.

**Roadmap**

Para alcanzar las 3 metas se propone:

1. **Levantar un portal corporativo:**
   1. Comprar dominio: a. Conseguir el dominio corporativo "http://www.bloocru.com". Esto se debe comprar, no hay muchas opciones, aunque en una primera instancia se puede conseguir un dominio gratuito, pero no corporativo, como "http://broocru.co.nr", por ejemplo.
   2. Conseguir hosting/server: Podemos utilizar el servicio Amazon la micro instancia gratuita para utilizarla como maquina principal para el proyecto, donde se mantendrá el portal corporativo o página web, además de un demo del producto.
   3. Levantar servicios: Si usamos Amazon podemos instalar algunos módulos bitnami para disponibilizar rápidamente los servicios web. Para correo corporativo podemos utilizar algún servicio gratuito como Zoho, la cual ofrece servicios gratuitos para empresas, libre para 3 usuarios: <https://mail.zoho.com/mailsignup.do?plan=free>
   4. Levantar portal Corporativo: Si completamos lo anterior podemos levantar una página de bienvenida rápida en una primera instancia, para pasar a un blog corporativo o un sitio web.
2. **Generar una maqueta/prototipo de la aplicación para facilitar la venta:**
   1. Después de completar lo anterior podemos realizar un diseño del producto, a bajo nivel, de forma que podamos determinar cómo vamos a cumplir con todos los requerimientos.
   2. Luego podemos diseñar una interfaz de pantalla, basado en los mockups que se poseen actualmente y considerando cómo funcionará el backend, de forma de conseguir una maqueta suficientemente consistente con el producto final.
   3. Una vez completada la primera versión de la maqueta se puede pasar a la primera versión del backend, para tener un primer prototipo funcional.
3. **Completar una primera versión funcional del producto**
   1. Una vez completado el prototipo, y pasando por ajustes en el diseño generados por el feedback que los usuarios nos entreguen, podemos pasar a una primera versión del producto.

**Diseño preliminar: Funcionalidades**

Éste diseño queda sujeto a revisión, pero la idea es proponer un camino para concretar un diseño a bajo nivel que nos permita generar una maqueta lo antes posible. Para generar éste diseño dividí el producto en las funcionalidades requeridas que pude identificar desde los mockups en balsamiq, donde pude identificar. Éstas características no deben ser mapeadas una a una a pantallas, si no que solo indican funcionalidades que la aplicación debe proveer y cuya información asociada debe ser registrada, éste diseño apunta a definir la mecánica para almacenar y recuperar ésta información.

1. Infraestructura: Más que una característica de la aplicación debemos partir diseñando la base del sistema, que plataforma utilizaremos, como base de desarrollo. La idea, como está definido anteriormente, es utilizar AWS en conjunto con algunos módulos bitnami. Todos los software o plataformas que se indican a continuación son compatibles con ésta arquitectura.
2. Dashboard/Analytics: Se requiere mostrar estadísticas de uso del sistema, de forma general y por módulos, además de información por usuario. Para registrar esta información se propone usar **Piwik**, ésta herramienta open source permite analizar tráfico y estadísticas del uso de un sitio, con una interfaz web interesante. Además posee una API, por lo que podemos usarlo para medir otras variables que nos interesen y generar reportes al respecto http://es.piwik.org/.
3. Geolocalización: Se requiere mostrar información georeferenciada en el portal, y asociar información a lugares. Esto lo podemos resolver utilizando la API de google para integrar el portal con **Google-Maps**, el cual es gratuito bajo ciertas condiciones, y de pago bajo cierto nivel de carga dentro de un portal.
4. Chat: Se requiere en el sistema generar un espacio de interacción directa entre dos usuarios, para lo cual podemos utilizar uno de los múltiples plugins disponibles en la web, en el camino debemos decidir si utilizamos alguno pagado, o ajustamos alguno gratuito. Aquí puedes ver algunos de los que revisé:
   1. <https://www.zopim.com/product>
   2. <http://livechat.comm100.com/>
   3. http://livehelp.livezilla.net/home/en/
5. Tips y Beneficios: Se requiere almacenar consejos, como si fuesen posteados en un muro, y se debe permitir la búsqueda de los post, y hacer comentarios. Para registrar esta información podemos utilizar phpBB, el cual es un foro open source, disponible en bitnami y escrito en PHP.
6. Eventos: Se requiere registrar mensajes registrando además una fecha y una posición geográfica. Para esto podemos usar phpBB también.
7. Destinos: El sistema requiere que se registre información sobre los destinos que se encontrarán disponibles para viajar, dado que esto es principalmente información estática, deberíamos registrarla directamente en una base de datos, mysql o postgresql son dos opciones disponibles gratuitamente en bitnami.
8. Información de los Viajes: Se requiere registrar la información de los viajes que se realizan, también asociar ésta información con la de los usuarios/tripulantes que viajarán y los destinos asociados. Para esto también podemos realizar nosotros mismos el registro directamente en alguna base de datos.
9. Cambios de Vuelos: Se requiere publicar la solicitud de cambios de vuelo, para esto podemos utilizar phpBB para hacer un post de los avisos y publicitarlos, además deberemos registrar manualmente en la base de datos los cambios efectivamente realizados, para futuras referencias.
10. Preguntas y Respuestas: El sistema debe permitir el registro de preguntas, postear respuestas y rankear los resultados hasta encontrar la mejor respuesta disponible, para esto podemos utilizar osqa, la cual es una plataforma open source y disponible en bitnami para el registro de preguntas. http://www.osqa.net/.
11. Perfil: Finalmente el sistema debe permitir el registro de los usuarios, junto con su información personal y preferencias, para facilitar que el sistema calcule información para ofrecerle a cada usuario. Para esto deberíamos registrar la información de cada usuario directamente en una base de datos, y relacionar ésta información con la información de acceso a los sistemas periféricos.

**Diseño preliminar: Integración**

Para concretar las funcionalidades anteriormente descritas se pretende utilizar directamente el software indicado anteriormente en una primera instancia, para tener un prototipo temprano, ya acercándonos a producción será necesario pasar a un segundo nivel de integración para mejorar el rendimiento del producto final.