

Prototipado y experiencia de usuario

Tabla de contenido

Prototipos.....	2
Pasos tentativos en el espacio de la solución	2
Distintos tipos	2
En el proceso de desarrollo	3
Experiencia del usuario	3
Principios	3
Caso.....	4
Sistema de reservas	4
Visión del producto	4
Definir conversaciones	4
Necesidad original.....	5
Requisitos descubiertos	5
Prototipado	6
Riesgos.....	6
Recomendaciones.....	6

Prototipos

Pasos tentativos en el espacio de la solución

Lo primero que debemos clarificar es, ¿qué es un prototipo? Un prototipo, en el contexto de la ingeniería de software, **es un paso tentativo en el espacio de la solución que nos ayuda a resolver incertidumbres en forma temprana.**

Usualmente construimos prototipos para:

- Aclarar, completar y validar los requisitos.
- Para explorar alternativas de diseño.
- Para desarrollar una parte del producto que pueda, posteriormente, evolucionar y transformarse en el producto final.

[V / F] Un prototipo es un paso tentativo en el espacio de la solución que ayuda a resolver incertidumbres en forma temprana.

Distintos tipos

Los prototipos se pueden clasificar de acuerdo a su uso futuro y a su alcance.

De acuerdo a su uso futuro, se pueden clasificar como:

- Los **descartables** se desechan, no evolucionan para convertirse en el producto final. Dentro de los descartables encontraremos los llamados **wireframes**, que son un tipo particular de prototipo descartable que se utiliza para clarificar requisitos, comprender mejor la arquitectura de la información y definir la posible navegación entre los diferentes espacios de interacción o interfaces de usuario.
- Los **evolutivos**, en cambio, sí evolucionan para convertirse en el producto final.

[V / F] Los *wireframes* son prototipos del tipo evolutivo.¹

Los prototipos también se pueden clasificar de acuerdo a su alcance:

- Un **Mock Up** es un prototipo de una parte del interfaz del usuario. También es llamado por esta razón, prototipo horizontal. No implementa comportamiento, pero lo sugiere. Muestra interfaces y también navegación.
- Una **PoC** (Proof of Concept) o prueba de concepto, en cambio, **es un prototipo vertical** que implementa una pequeña porción de la aplicación desde la interfaz del usuario hasta la última capa técnica con el propósito de determinar la viabilidad técnica de algún aspecto de implementación. En las metodologías ágiles suelen ser llamados Spike.

[V / F] Una PoC es conocida como un prototipo horizontal ya que no implementa comportamiento pero lo sugiere.²

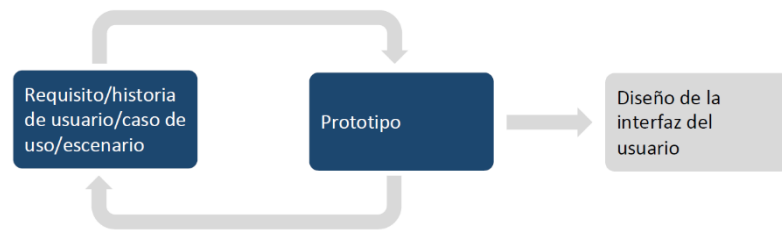
¹ Los wireframes son prototipos del tipo descartable.

² Una PoC es conocida como un prototipo vertical ya que implementa una pequeña porción de la aplicación.

En el proceso de desarrollo

¿Cómo se pueden utilizar los prototipos en el ciclo de vida? Cada requisito, cada historia, cada caso de uso, cada escenario describe una tarea que el usuario realizará con el sistema. A partir de aquí se identifica un diálogo que puede representarse mediante un gráfico llamado **mapa de diálogo**. Este diagrama

permite representar las distintas interfaces o espacios de interacción que participan en el diálogo entre el usuario y el producto. Cada interfaz o espacio de interacción será posteriormente representado mediante un prototipo de interfaz con todos los elementos de datos que hagan falta. Con este prototipo podremos validar y refinar los requisitos. El análisis no se debe centrar en los aspectos visuales del prototipo, sino más bien en la información que el usuario debe ingresar y que el sistema debe mostrar. Es muy fácil caer en discusiones prematuras acerca del diseño visual, tema que debemos atacar posteriormente.



[V / F] El diagrama conocido como mapa de diálogo permite representar las distintas interfaces que participan entre el usuario y el producto.

Lo que siempre hay que tener presente al preparar los prototipos y posteriormente al diseñar las interfaces, es que el usuario mantiene un diálogo o conversación con el producto con el propósito de realizar tareas que le permitan cumplir con sus objetivos. Como toda conversación, la interfaz del usuario debe ser amable y educada.

Experiencia del usuario

La comunicación entre personas tiende a ser natural, emplea un lenguaje claro, está orientada a conseguir objetivos, sigue un flujo natural y suele ser concisa y simple. Por el contrario la comunicación con productos tecnológicos suele ser poco natural, se suele emplear un lenguaje técnico, es mecánica, suele seguir la forma en la que trabaja el código y es complejo.

[V / F] Es clave que el análisis con prototipos se centre en los aspectos visuales y en la información que el sistema debe mostrar.

Principios

Lo que tenemos que tener en cuenta para definir una buena experiencia del usuario con el producto son los siguientes principios:

- 1º) La interfaz del usuario es comunicación.
- 2º) Las tareas deben explicarse clara y concisamente, como si fuera en persona.
- 3º) Todos los elementos de la interfaz del usuario pueden ser evaluados por qué comunica y cómo lo hace.
- 4º) Hay que ser amable, respetuoso e inteligente.
- 5º) Si la interfaz del usuario se siente como una comunicación natural, profesional y amigable, probablemente sea por su buen diseño.

Caso

Sistema de reservas

Veamos un ejemplo: Debemos preparar el prototipo de un sistema de reserva de vuelos. Nuestro stakeholder nos hace este planteo: “Necesitamos una aplicación que le permita a nuestros clientes comprar pasajes para viajes aéreos”.

Visión del producto

Adicionalmente, sabemos que:

- La compañía DK Travel es una agencia de viajes global que quiere ofrecer sus servicios en línea.
- El objetivo de DK Travel son los viajeros de negocios y los turistas que buscan grandes ofertas.
- Su nuevo sitio de viajes llamado dk-travel.com debe reflejar estas fortalezas, al ayudar a los viajeros de negocios y de placer, a comprar los pasajes correctos, al mejor precio, con un mínimo esfuerzo.
- El sitio será gratuito para los usuarios DK Travel va a cobrar una comisión a las aerolíneas cuando los clientes compren un ticket.

Probablemente nuestro primer intento de prototipo sea algo parecido a esto:

Claramente influenciado por el modelo de dominio o el modelo de datos. Queremos buscar un vuelo entre los aeropuertos, para una fecha determinada, con una determinada cantidad de tramos, para una cantidad de personas de diferentes edades, y necesitamos tickets de una clase determinada. Claramente no estamos ante una conversación demasiado cordial. Hay mucha información para ingresar y no sabemos muy bien qué debemos hacer.

A Web Page
http://dk-travel.com

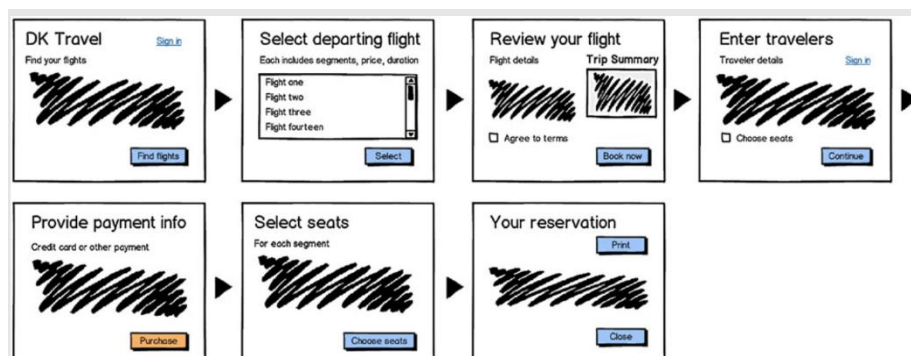
Flight Options

Flight 1: Departure airport: Airport code
Departing: mm/dd/yy
Arrival airport: Airport code
Segments: 1
Travelers: 1
Traveler ages: 0 0 0 0 0
Ticket class: All
☐ Non-stop only
☐ Refundable only

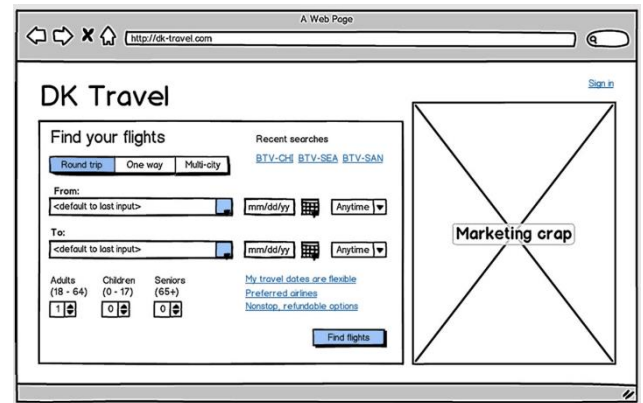
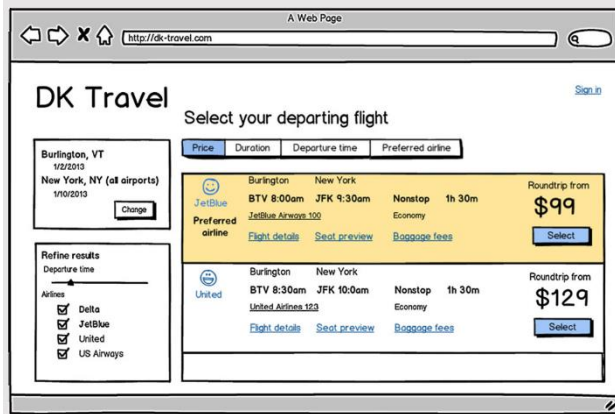
Definir conversaciones

Si indagáramos un poco más, descubriríamos que reservar un vuelo es un proceso iterativo. Así sabríamos que algunos usuarios podrán tener determinado su destino, pero pueden ser flexibles con respecto a las fechas y horarios. Otro viajero puede ser que ni siquiera sepa exactamente qué es lo que quieren y necesiten explorar alternativas. En todos los casos es evidente que hay un diálogo, una conversación, que le debe permitir tomar una decisión y reservar su vuelo.

Con esta información podemos comenzar a elaborar una versión mejorada de nuestro prototipo, más orientada al flujo de la tarea y no tanto al ingreso de datos.



Primero buscaremos el vuelo. Le daremos la posibilidad al usuario de elegir el tipo de viaje (ida, ida y vuelta, múltiples destinos); los aeropuertos de salida y llegada, la fecha de salida o llegada (alternativamente ninguna); y la cantidad de pasajeros por rango de edad. En esta propuesta, la aplicación permitirá recuperar las últimas búsquedas realizadas con anterioridad.



Luego de ingresar estos datos, el sistema mostrará los vuelos que cumplan con los criterios ingresados en el paso anterior. Podrá consultar los detalles del vuelo, la disposición de los asientos, los costos del equipaje. Podrá ordenar los resultados por precio del pasaje, por duración del vuelo, por fecha y hora de partida o de acuerdo a su aerolínea preferida. El usuario podrá seleccionar el vuelo deseado y proceder luego a seleccionar los asientos y realizar el pago.

Necesidad original

Lo que se puede ver con este ejemplo, es que el prototipo, luego de la revisión con el usuario, nos ha permitido identificar varios requisitos derivados de la necesidad inicial: “Necesitamos una aplicación que le permita a nuestros clientes comprar pasajes para viajes aéreos”.

Requisitos descubiertos

- El usuario debe poder:
 - Consultar vuelos disponibles, indicando tipo de vuelo, aeropuertos de salida y de llegada, fechas tentativas de salida o llegada, cantidad de pasajeros por franja etaria.
 - Indicar cuáles son las aerolíneas preferidas y si sus fechas de viaje son flexibles o no.
 - Consultar las políticas de reintegro.
- Para los vuelos que cumplan los criterios de búsqueda ingresados, el sistema deberá informar
 - La aerolínea e identificación del vuelo.
 - Las ciudades y aeropuertos de partida y llegada.
 - Los horarios de partida y llegada, la duración del vuelo, las escalas.
 - Los detalles del vuelo, la disposición de asientos de acuerdo al modelo de avión, los costos del equipaje.
- Adicionalmente, ofrecerá la posibilidad de ordenar los vuelos por precio, duración, hora de salida y aerolínea preferida.
- Con los resultados obtenidos, el usuario debe poder:
 - Ordenarlos por precio, duración, hora de salida, aerolínea.
 - Refinar la búsqueda.
 - Cambiar la búsqueda.
 - Consultar los detalles de los vuelos, la disposición de los asientos, los precios de los equipajes.
 - Seleccionar un vuelo que se adecúe a sus preferencias.

- Con el vuelo seleccionado, el usuario debe poder:
 - Seleccionar los asientos.
 - Completar los datos de los pasajeros (pasaporte/DNI, nombre y apellido, edad).
 - Realizar el pago mediante su tarjeta de crédito o débito.

Hay que tener en cuenta que para definir una buena experiencia de usuario los siguientes principios

- ☒ **Las tareas deben explicarse clara y concisamente**
- ☐ Algunos elementos de la interfaz de usuario sólo son evaluados por qué comunican³
- ☒ **La interfaz del usuario es comunicación**

Prototipado

Riesgos

Si bien es muy útil, el prototipado no está exento de riesgos. Entre los más importantes tenemos los siguientes:

- ❖ Presión para liberar el prototipo.
- ❖ Foco en detalles del diseño visual. No es el momento de discutir acerca de íconos, colores, fonts. Eso es algo que deberíamos dejar para más adelante.
- ❖ Expectativas de performance.
- ❖ Y puede ser también que implique un esfuerzo demasiado grande el desarrollo de los prototipos.

Recomendaciones

Para cerrar entonces, algunas recomendaciones:

- ★ Incluir el prototipado en el plan de trabajo. El prototipo va implicar un esfuerzo, un tiempo, un costo que hay que incluir en el plan de trabajo.
- ★ Hay que clarificar el propósito del prototipo. Tiene que quedar claro para qué lo estamos construyendo.
- ★ Hay que planear el desarrollo de múltiples prototipos para distintos aspectos, para distintas partes del sistema.
- ★ Tenemos que estar en condiciones de poder desarrollar rápidamente prototipos, prototipos que sean descartables y baratos de construir.
- ★ Tenemos que poner el foco en prototipar aspectos no entendidos o que impliquen cierto grado de riesgo.
- ★ Tenemos que usar datos creíbles, eso va a facilitar muchísimo la revisión con los usuarios.
- ★ Recordar que no tenemos que reemplazar los requisitos con el prototipo. Los prototipos son un mecanismo para validar requisitos, para analizarlos, para encontrar nuevos requisitos, pero no son requisitos en sí mismos.

[V / F] Realizar prototipos involucra un esfuerzo que debe ser contemplado en la planificación.

³ TODOS los elementos de la interfaz del usuario pueden ser evaluados por qué comunican