

# INFORME FUNCIONAL TP1

Interacción Hombre-Computadora



*Best*  
Flight

---

*Grabina – Grethe - Burgos*

---

ITBA  
1C 2017

## Tabla de contenidos

Introducción.....	2
Capturas de pantalla.....	3
Página principal.....	3
Listado de vuelos .....	4
Formularios de información.....	6
Ofertas .....	8
Estados de vuelos.....	9
Confirmación.....	9
Implementación.....	11
Estadísticas provistas por GitHub: .....	11
Estructura.....	12
Estilo.....	12
Comportamiento .....	12
Instalación.....	13
Requisitos.....	13
Navegadores Soportados.....	13
Guía para la instalación.....	13
Conclusión.....	14

## Introducción

En esta segunda y definitiva entrega, sobre el trabajo practico grupal número uno, en el que se evalúan entre otras cosas y en líneas generales, nuestras capacidades de desarrollo una interfaz de usuario eficiente y optima en cuanto a criterios de usabilidad en un entorno de desarrollo web, mediante la utilización de tecnologías aprendidas en clase como HTML, CSS, JavaScript y otras aprendidas por cuenta propia como Bootstrap y API's de terceros, se plantea la solución final propuesta por el grupo, basados en conceptos teóricos de la materia y fundamentando criterios elegidos durante su desarrollo.

## Capturas de pantalla

Se aclara que en este informe no se adjunta la totalidad de las pantallas, únicamente las relevantes para fundamentar las decisiones de usabilidad.

### Página principal



Imagen 1 – Página principal.

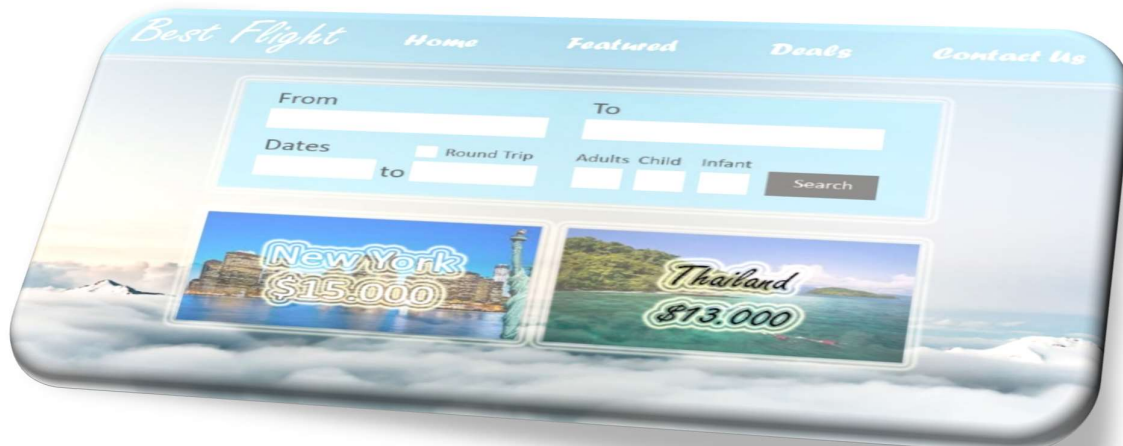


Imagen 2-Prototipo página principal.

Siguiendo las recomendaciones de la cátedra del primer informe, se arreglaron los problemas de coherencia y capitalización en todas las páginas.

Siguiendo los consejos de la lectura 17, en la parte superior de la página ubicamos nuestro logo, para que el usuario pueda identificar el objetivo de la página. Además, Intentamos que la cantidad de texto sea mínima y que la pagina entre en una pantalla sola, para evitar el scrolling.

En cuanto a la barra superior, se hicieron algunas encuestas y la mayoría de los usuarios no reconocía la diferencia entre featured y deals, por lo que se decidió juntar las dos páginas en una.

Dentro del cuadro de búsqueda se movió el round al cotado para hacerlo más visible. Además, seteamos el input box de adult en 1 y los demás en 0, dado que nuestras encuestas dieron que la mayoría de las personas busca el precio de una persona para comparar con otras páginas.

Todos los datos de esta pagina son validados por la pagina. No se permiten vuelos desde una misma ciudad y en el caso de buscar vuelos ida y vuelta, la fecha de ida debe ser anterior a la de vuelta.

Por ultimo, al cargar la pagina en la parte superior se encuentra un cartel que avisa que no se encuentra javascript instalado, vital para el funcionamiento de la pagina. Este cartel es removido por javascript apenas termina de cargar. Este chequeo lo hacemos solo en esta pagina, dado que es la primera al que el usuario ingresa.

## Listado de vuelos

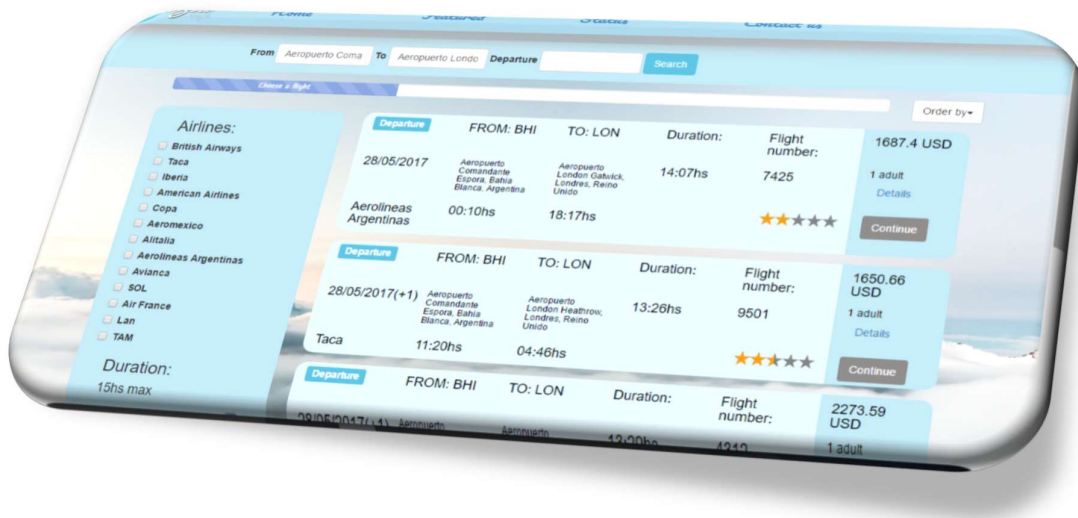


Imagen 3 - Listado de vuelos disponibles en base a la búsqueda anterior



Imagen 4: prototipo del listado de vuelos

La página de búsqueda de vuelos es definitivamente la que más cambio con respecto a nuestro prototipo.

Primero en la parte superior de búsqueda de vuelos, se decidió remover varias opciones. La cantidad de personas siempre va a ser la misma dado que, por lo que se observó en las evaluaciones de usabilidad, los usuarios nunca lo cambiaban. Mientras que, en cuanto a las fechas, si la persona eligió solo ida no va a poder realizar una búsqueda ida y vuelta. Para esto tendría que volver al home y realizar la búsqueda de ahí.

En cuanto a los filtros, se removió el de conexiones ya que no fue implementado. Se dejó el de aerolíneas y cambiamos el de precio por uno de tiempo con el mismo estilo.

Dentro de la búsqueda de vuelos, se agregó toda la información faltante que nos recomendó la catedra en la primera entrega, como por ejemplo el día de llegada, el tiempo de vuelo, etc. Además, cuando la búsqueda es ida y vuelta, se reemplazó el cuadro de ida y vuelta en la misma página. Ahora, primero se despliega una página para buscar el vuelo de ida y al seleccionar se cargan los vuelos de vuelta. Decidimos realizarlo así porque nos parecía más fácil de implementar y más entendible para el usuario.

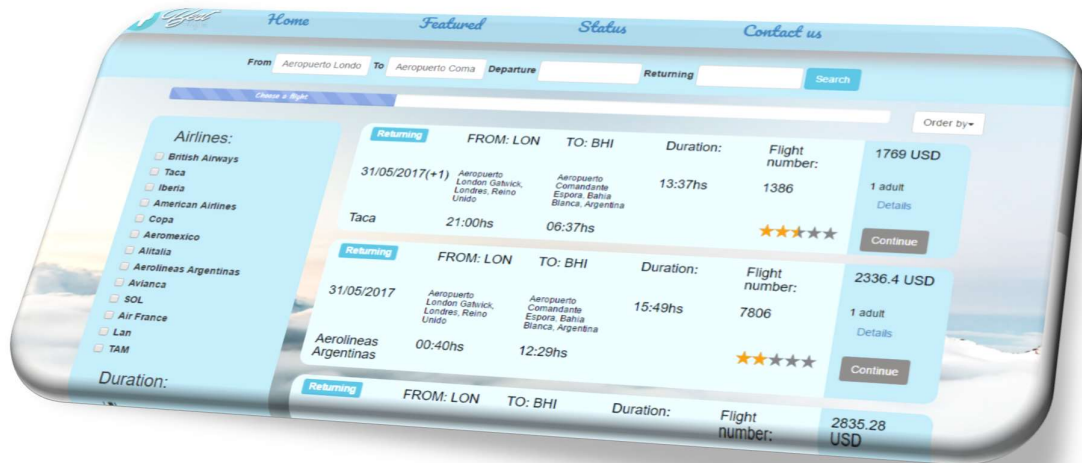


Imagen 5- Elección de vuelo de vuelta.

Por último, se agregó la opción de ordenar los vuelos según 3 posibilidades: por precio de menor a mayor, por duración y por aerolínea. Este botón se encuentra a la derecha de la barra de progreso.

## Formularios de información

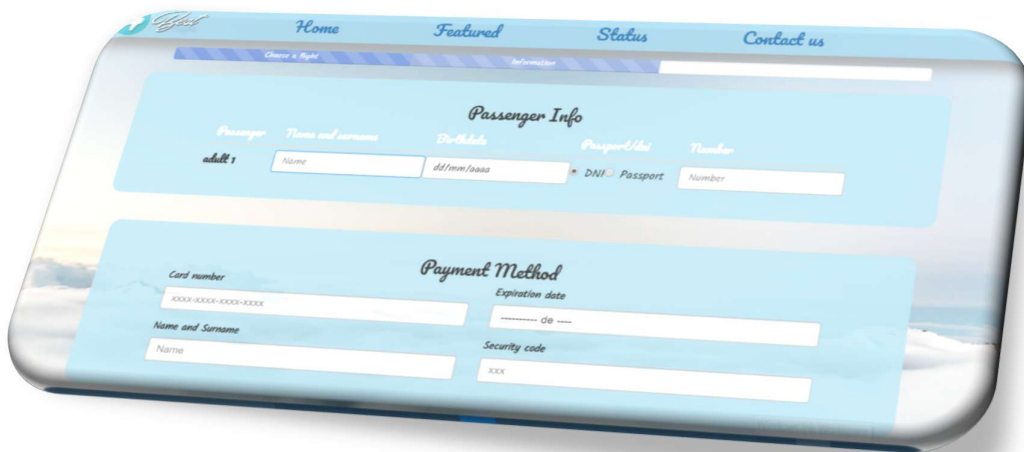


Imagen 6- Formularios de información.

La página de información decidimos rehacerla completamente, por lo que no incluimos el prototipo en este trabajo, ya que no cumplía con todas las exigencias necesarias en cuanto a formularios de información.

En la parte superior de la página se encuentra la información de los usuarios. Se carga la cantidad de pasajeros según indico el usuario y se aclara en la primera columna si son adultos, niños o bebés. La fecha de nacimiento es validada por la página, como por ejemplo si se ingresa un niño

con una edad superior a 11 años, aparece un error al intentar confirmar. Por lo tanto, si el usuario se confundió la página le va a aclarar que la fecha de nacimiento es invalida y tendrá que volver a realizar la compra.

Más abajo se encuentra el cuadro de pago. En esta se pide la información básica para poder realizar la compra. Como recomendó la catedra, lo primero que se pide el número de tarjeta de crédito. Todos estos datos están validados por la api. La tarjeta y toda su información es validada por la api. En el último lugar dentro de esta tabla se encuentran las cuotas. Estas son dadas por la api. En caso de que sea un vuelo de ida y vuelta, le pedimos a la api la cantidad de cuotas disponibles tanto para la ida como para la vuelta y solo se muestran las que estén disponibles en los dos vuelos.

The image shows a 3D mockup of a light blue payment form with rounded corners. The form is divided into two main sections: "Billing information" and "Contact". The "Billing information" section contains fields for Country, City, State, Zip Code, Address (with a placeholder "myAddress 1234"), Floor (Optional), and Apartment (Optional). The "Contact" section contains fields for Email (with a placeholder "myemail@email.com") and Phone Number.

Imagen 7- Parte inferior del payment.

En los últimos dos recuadros se pide la información del usuario que va a realizar la compra. El país y la ciudad son válidas por la página, por lo solo es posible comprar en los lugares provisto por la api. El estado solo se valida la cantidad de caracteres máximos, al igual que el zip code que además se le valida que solo sean números.



## Ofertas



Imagen 8 – Ofertas de vuelos.

En la parte superior de la página se encuentra un mapa, provisto por la api de Google Maps, la cual carga la ubicación de todos los destinos en oferta para que el usuario pueda ver de manera intuitiva e interactiva los potenciales destinos entre los que podrá elegir.

Abajo se encuentra un select button, el cual el usuario debe seleccionar desde donde viaja. Al seleccionar carga todos los destinos en oferta. En la primera columna se carga una imagen alusiva, provista por la api de Flickr. En la segunda columna hay el nombre de la ciudad con el link a la búsqueda de vuelos y en las últimas dos columnas el país de la ciudad destino y el precio mínimo que sale un viaje para un adulto, toda la información pertinente pedida por el enunciado de la catedra.

Hay que aclarar que nuestra página, al seleccionar el destino, redirige al usuario a la página de búsqueda de vuelos. Esta búsqueda es por default para un adulto, basada en la estadística que un adulto solo es el mayor porcentaje de búsquedas en sistemas de reservación de vuelos en el mundo.

## Estados de vuelos

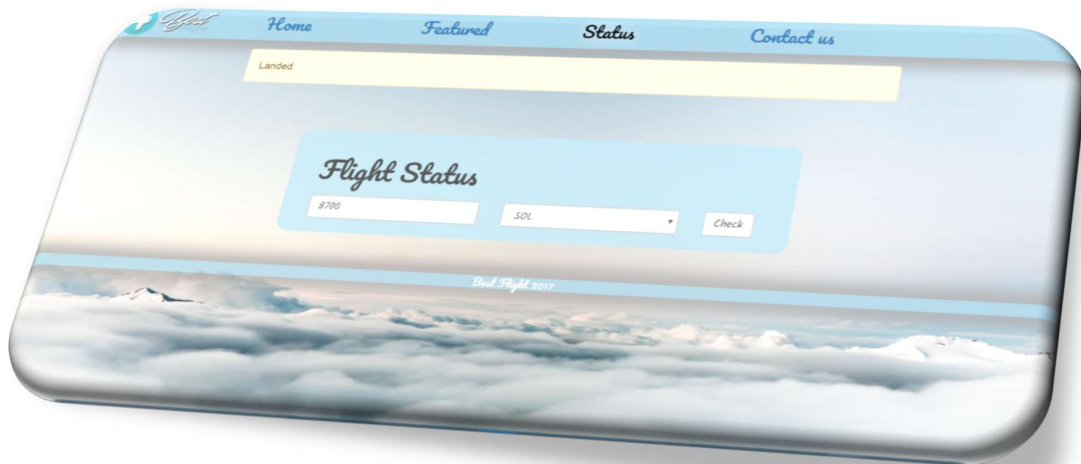


Imagen 9 – Estados de vuelos.

La página del status de vuelo, intentamos hacerla lo más simple posible. Se selecciona el vuelo, la aerolínea y al apretar el botón de “check” salta desde la parte superior el estado del vuelo. En caso de que el vuelo sea incorrecto se lanza un mensaje de error correspondiente en cada caso.

## Confirmación

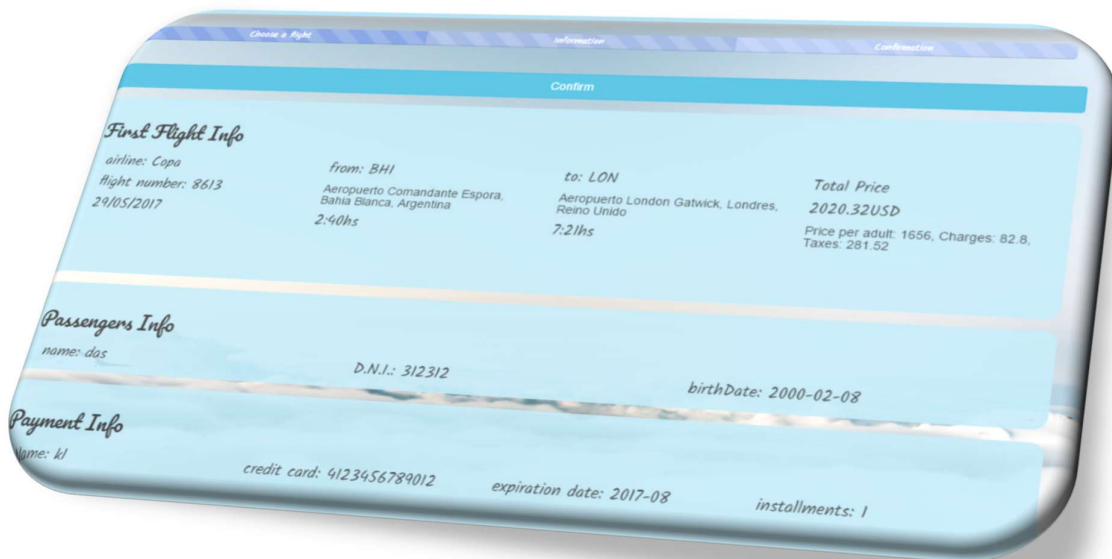


Imagen 10- Pagina de confirmación.

Por último, la página de confirmación muestra toda la información relevante en cuanto al vuelo que se va a comprar. Se encuentra en la parte superior un botón de confirm, ya que, por lo visto en los test de usabilidad, la mayoría de los usuarios no pierde el tiempo leyendo esta página. Al apretar el botón te avisa si se pudo realizar la reserva. En caso afirmativo desaparece el botón de confirm, para evitar errores de confirmar un mismo vuelo varias veces.

Las capturas de pantalla correspondiente a la página de contacto no se incluyó en el informe debido a que no es relevante en cuanto al análisis que se realiza del sistema.

## Implementación

El equipo fue trabajando en conjunto sin separaciones claras en cuanto a archivos, mediante la tecnología GitHub creando un repositorio público en <https://github.com/mgrabina/HCIWEB>.

En cuanto a las tres categorías, corresponde aclarar que a medida que el equipo fue aprendiendo las tecnologías y descubriendo las capacidades de las mismas, se fue agregando contenido, estilo y comportamiento extra que no se ve reflejado en los prototipos iniciales, ya que para aquel entonces no se sabía con exactitud los límites y posibilidades provistos por los lenguajes que finalmente utilizamos. Todas estas mejoras fueron pensadas en base a la teoría aprendida en clase y con el fin de brindar una mejor experiencia al usuario.

Además, utilizamos Bootstrap y JQuery.

Estadísticas provistas por GitHub:



*Gráfico: Contribuciones de código en función del tiempo*

## Estructura

En cuanto a la estructura del sistema, decidimos que se mantenga a lo largo de toda la experiencia que tiene el usuario con el mismo, es por esto por lo que se mantiene la forma en la que se presenta el header, el menú superior, el footer y la ubicación de los contenidos.

Basamos la estructura en el “grid sistem” de Bootstrap para facilitar la implementación y mejorar su diseño responsivo.

## Estilo

Una de las decisiones que se llevaron a cabo relacionadas al estilo de las paginas, fue centrar en un archivo, aquellas propiedades que se repiten en todas las pantallas del sistema, como son las de los header, footer, body, entre otras, su nombre es general.css, se encuentra en la carpeta junto a las demás hojas de estilos y es referenciada desde todos los documentos HTML.

Aquí también se aplicó clases de Bootstrap.

## Comportamiento

Se asume que siempre el usuario ingresa al sistema por el índice, por lo cual se realiza el testeo de JavaScript únicamente en esa página.

En cuanto al tratamiento de errores y mensajes de alerta se decidió manejarlos de manera consistente durante todo el sistema y que sean lo menos intrusivos posibles hacia la experiencia del flujo natural de la reservación que posee el usuario, es por esto, que aparecen en forma de carteles sin interrumpir el uso (a diferencia de los carteles de alerta naturales de los navegadores), al lado de los botones que potencialmente lo ejecutaron, donde será lo más probable que se encuentre la vista del usuario.

Se utilizan API's de terceros mediante JavaScript para agregar foto alusivas y mapas al comportamiento del sistema, provistas por Flickr y Google Maps respectivamente.

## Instalación

### Requisitos

- Conexión a internet con acceso a los servidores del ITBA
  - Evitar red Wifi ITBA-Segura
- JavaScript preinstalado
- Tener memoria disponible tanto en el almacenamiento local como de sesión del navegador.

### Navegadores Soportados

- Google Chrome
- Opera
- Edge
  - Requiere Web Server

### Guía para la instalación

1. Descargar la carpeta del sistema con los archivos.
2. Incluir en la carpeta de su Web Server e iniciarlo.
3. Abrir el archivo index.html en el navegador.

## Conclusión

Dado por finalizado nuestro primer proyecto Web, concluimos que las tecnologías que se usaron son potenciables para hacer gran cantidad y calidad de sistemas diferentes, en este caso, un sistema de reserva de vuelos.

Creemos que con más tiempo disponible se pudiese haber mejorado el sistema en relación a sus tres componentes: estructura, presentación y comportamiento. Incluyendo también nuevas API's de terceros como fue el caso de Google Maps.

Sin embargo, entendemos que la tendencia a agregar comportamiento para mejorar la experiencia de usuario puede ser contraproducente, ya que la suma de muchas peticiones AJAX, y posterior análisis de datos puede llevar a la página a demorar en su carga y generar un efecto recíproco en la búsqueda de una mejor UX.

Por último, habiendo cumplido con todos los requisitos funcionales y no funcionales del enunciado, damos por finalizado el primer trabajo práctico, si así lo considera la cátedra.