**Informe Académico:**

**.Desarrolladores:**

.David Cabrera García C-312

.Abel Molina Sánchez C-311

**.Presentación:**

Se presenta la aplicación web **HV-HayViajes!**, la cual constituye una plataforma intermediaria entre agencias de turismo y clientes potenciales. Este trabajo surgió resultado de la necesidad por parte del cliente de ofrecer un servicio práctico a la vez que accesible a los futuros usuarios del sistema. La aplicación está enfocada principalmente a dos tipos de usuarios que serán los que interactuarán con ella constántemente: los turistas y los mánager de agencias. Los primeros van a ser capaces de realizar reservaciones de todos y cada uno de los servicios que oferten las agencias. En un primer momento estos servicios están constituidos por los paquetes, los hoteles y las excursiones, pudiendo realizarse reservaciones combinadas de estos últimos.

**.Requerimientos:**

Con el desarrollo del software de **HV-HayViajes!** Teníamos que ser capaces de cumplir y dar respuesta a un grupo de funcionalidades planteadas por el cliente, y siempre abierto a la extensión de las mismas, así como a modificaciones que se ajustaran en mayor o menor medida a los casos de uso de la interfaz.

**HV-HayViajes!** Como plataforma intermediaria entre ofertadores y consumidores, no podía por esa naturaleza dejar de tener semejanza con otras propuestas similares en el mercado, como la plataforma de concertación de vuelos IDreams por solo citar un ejemplo.

Por tanto era una característica indispensable a implementar en nuestro sistema una navegación amigable para el usuario a través de las ofertas y los servicios, al fin y al cabo es un sitio para promover el consumo.

A su vez debíamos ser capaces de dotar a los distintos usuarios de algunos criterios de búsqueda especificados por el cliente y que constituían un requerimiento de nuestro sistema.

Dimos cumplimiento a las funcionalidades especificadas, y buscamos la mejor forma de adaptar cada una dentro del contexto del flujo de trabajo de la web, de forma tal que fueran una respuesta natural del sistema y una característica de provecho para el usuario.

Dentro de estas adaptaciones podemos encontrar la funcionalidad de listar por agencia los hoteles que incluyen sus paquetes, la cual es una funcionalidad a la que damos respuesta de forma natural a través de submenúes contextuales que se desprenden de las mismas tablas de los paquetes, pero que a su vez explicitamos agregando un enlace que permitiera mostrar específicamente esta característica. Dicho enlace aparece dentro de la página de forma no intrusiva con lo cual el usuario le dará uso de forma natural si requiere una vista diferente de dichos datos.

Otra adaptación dentro del funcionamiento, y que fue implementada bajo acuerdo previo con el cliente, fue el cambio de la funcionalidad que pedía listar a los turistas que viajaron a Cuba con una agencia. Debido a las características de los datos del sistema y la información con la que se contaba, dicha funcionalidad podría requerir cambios en la estructura del sistema así como la necesidad de incorporar datos que no existían de forma previa. De esta forma fue planteado al cliente, el cual se había percatado de dicha situación y por lo tanto se adaptó la nueva funcionalidad de listar por las agencias a aquellos turistas que hubieran realizado al menos una reservación, ya sea de paquete o individual a través de dicha agencia.

A su vez se incorporaron funcionalidades estadísticas para los managers de agencias, permitiendo a estos analizar el comportamiento de sus ventas y sus características dentro del mercado de **HV-HayViajes!.**

**.Análisis de la solución:**

Para el desarrollo de la solución se empleó el framework de Ptython, Django, el cual está estructurado bajo un patrón MVT, modelo, vista, template(plantilla). El manejo de datos y la comunicación con la base de datos del sistema se realiza utilizando el ORM integrado de Django, que permite trabajar los modelos de la Bd, como clases de python, lo cual hace mucho más intuitivo el manejo de la información.

La base de datos empleada en el sistema es sqlite3, que viene integrada directamente con django(python) con lo cual no hubo que configurar ningún manejador externo.

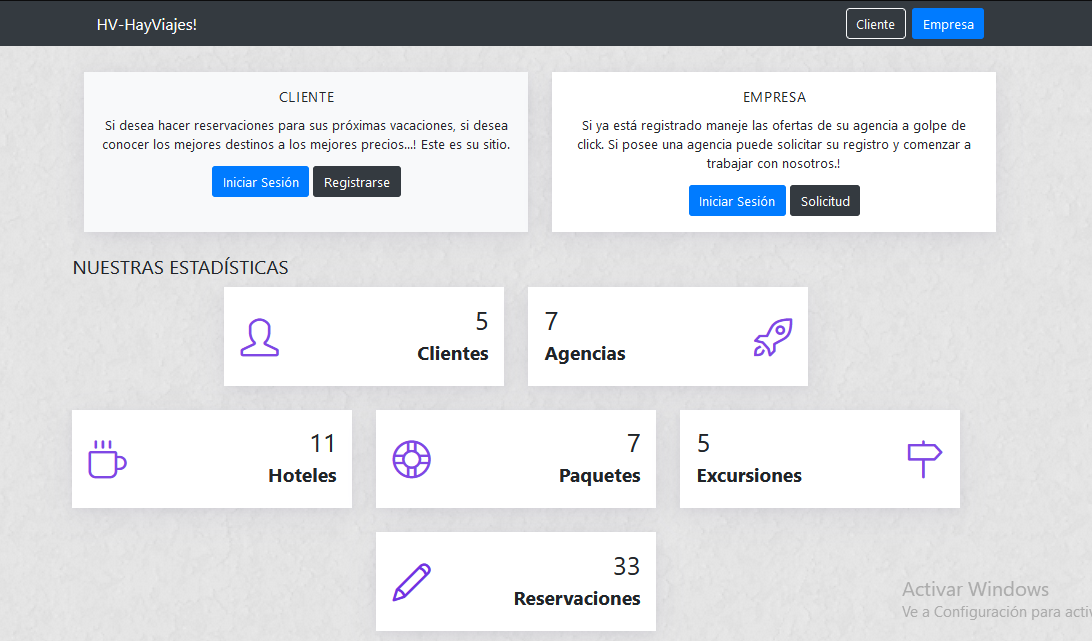
Para el desarrollo del la interfaz gráfica de la aplicación se agregaron los módulos compilados de boostrap4, jquery y dataTable principalmente, que son los que predominan en las páginas del sitio. Boostrap4, es un framework originalmente creado por Twitter, que permite implementar interfaces web personalizadas con CSS y JavaScript, y hacerlas “responsive”, esto es, adaptable a los distintos dispositvos. Además incluye una serie de estilos de personalización que hacen la interfaz mucho más amigable para el usuario. Jquery, es una librería de código abierto de JavaScript que simplifica mucho el trabajo con el lenguaje. Facilita el uso de selectores, así como también las peticiones ajax al servidor que es uno de los principales usos que le damos dentro de la aplicación. A su vez, dataTable es una extensión de jquery que como su nombre indica, se especializa en el trabajo con tablas, haciendo estas mucho más personalizables y permitiendo integrar en las mismas distintas características como los bucsadores generales y por columnas, así como el paginado y la adición de scrolls. Otra ventaja es que tiene una fácil integración con boostrap4.

Django, casi por obligación impulsa a cumplir con el principio DRY(Don't Repeat Yourself), ya que su API ofrece muchas características que hacen que el desarrollador no tenga que escribir código repetitivo para la parte del renderizado, y pueda centrarse principalmente en el manejo de la lógica. Esto es precisamente lo que sucede con los formularios, que son parte fundamental dentro de la vida de la aplicación ya que principalmente a través de los formularios es que ocurre el intercambio de información entre el cliente y el servidor. Como planteamos anteriormente Django es capaz de realizar de forma automática el rederizado de formularios creados heredando de su módulo forms. Además incluye validación por defecto de los campos a través de sus widgets de recepción de entrada. Para personalizar el renderizado de los formularios y hacerlos visualmente más atractivos al usuario intalamos los módulos de django: django-crispy-forms y django-widget-tweaks, que son desarrollados para ese fin y el resultado es de mucha calidad.

Para el trabajo con un selector de países personalizado, integramos el módulo django-countries, que ofrece un widget personalizado, y un campo field también personalizado y específico para la selección de países.

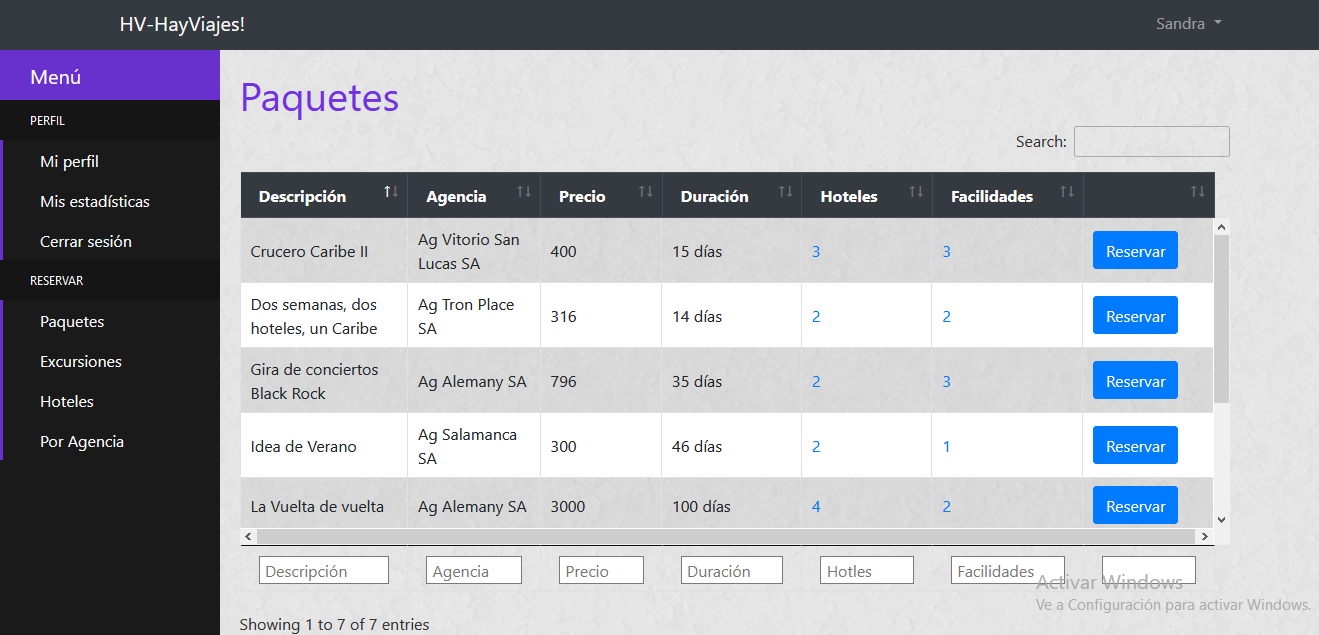
**.Resultados:**

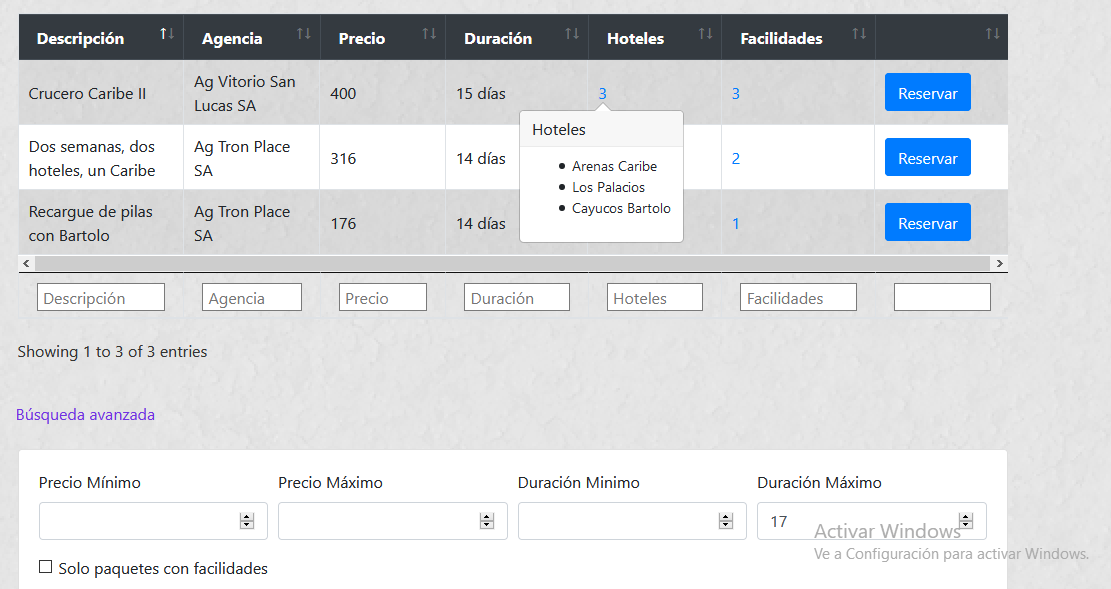
La aplicación está destinada a tres tipos de usuarios, los usuarios administrativos del sitio, que serán quienes tengan el control de la empresa, dando de altas a las agencias que deseen unirse a sitio y monitorizando las inclusiones de nuevos hoteles y/o excursiones. Para cualquier usuario la página principal de **HV-HayViajes!** se presenta de la siguiente forma:



Para aquellos usuarios que ya tengan una cuenta dentro del sitio, la página de los login los llevara a sus respectivos dashboards.

Un cliente natural, dígase un turista en busca de unas buenas vacaciones, se encontrará una interfaz sobria e intuitiva para elegir su mejor oferta.

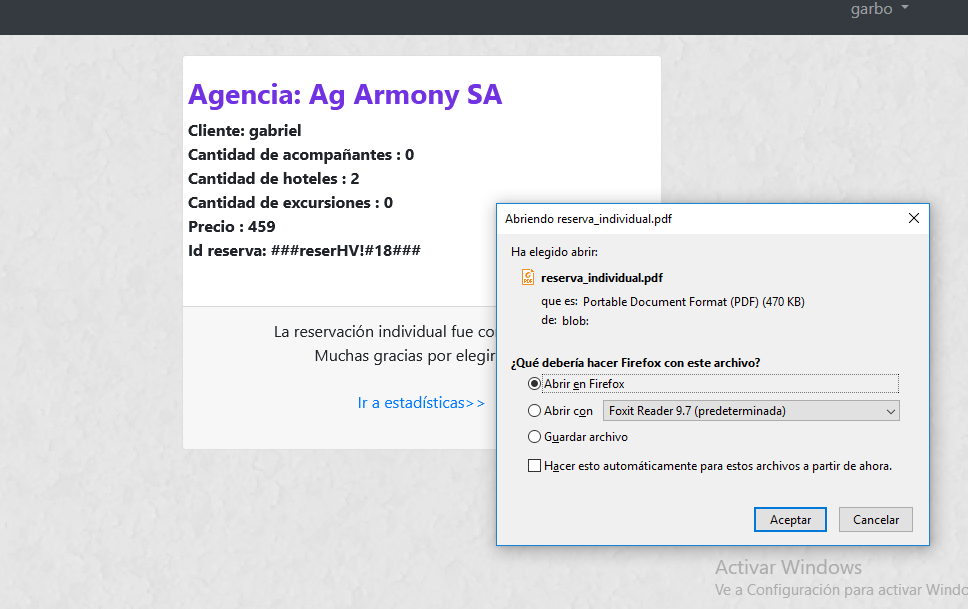




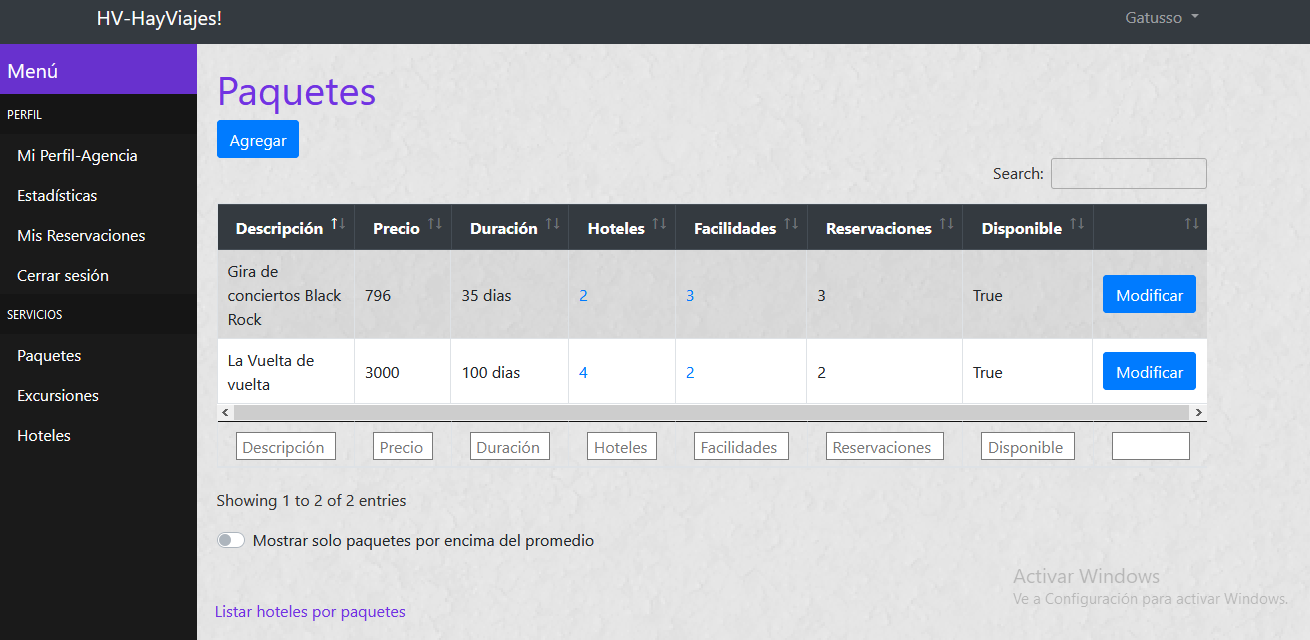
Estas interfaces permiten a los usuarios manejarse de forma intuitiva dentro del sistema. A su vez cuando un usuario completa una reservación del tipo que sea, recibe un ticket de reservación para su seguridad.

Del mismo modo en la sección de estadísticas puede llevar un control de su historial dentro del sitio y sus gastos en reservaciones.

Se inlcuyó la funcionalidad de reservar individualemente una único hotel, o una única excursión, que en si constituyen reservaciones individuales ya, pero le dan la posibilidad al usuario de tener más opciones de encontrar rápidamente lo que busca.



Los mánagers van a poder agregar y/o modificar sus paquetes de oferta, así como entablar nuevas relaciones con hoteles y/o excursiones disponibles de nuestro sitio, quedando en manos de la administración cualquier nueva incorporación a la empresa.



**.Conclusiones y recomendaciones:**

Como desarrolladores podemos decir que estamos satisfechos con el producto conseguido aunque siempre queda la sensación que puede ser mejorable y más extensible. A medida del que avanzaba el desarrollo en nivel de productividad fue aumentando en relación al dominio de las tecnologías las cuales eran todas nuevas para nosotros.

Luego de habernos enfrentado al problema y contrastado opiniones hasta con el mismo cliente en algunos momentos, consideramos que la presentación de este proyecto(el pdf explicativo) deja que desear. Y además consideramos que es un proyecto al que se le puede sacar mucho partido y un planteamiento más robusto y específico de lo que se desea realmente como producto final puede ser muy bueno.

**.Bibliografía:**

Para dar los primeros pasos en django nos apoyamos en la documentación oficial, así como otros blogs de desarrolladores que fueron muy instructivos como es el caso del blog del web developer **Vitor Freitas**.

-Documentación de Django 3.2: <https://docs.djangoproject.com/en/3.2/>

- Mozila developer: https://developer.mozilla.org/es/docs/Learn/Server-side/Django

- Blog de **Vitor Freitas**: https://simpleisbetterthancomplex.com/series/beginners-guide/1.11/

Para el inicio con JavaScript, ajax y jquery:

- <https://developer.mozilla.org/es/docs/Web/JavaScript/A_re-introduction_to_JavaScript>

- http://www.falconmasters.com/jquery/peticiones-ajax-jquery/

También utilizamos las páginas oficiales de boostrap4 y dataTable:

- <https://datatables.net/examples/>

- https://getbootstrap.com/docs/4.0/getting-started/introduction/