

## ANEXO EXAMEN DE CERTIFICACIÓN

Plan de Estudio	Desarrollo Aplicaciones Fullstack Python
Anexo	Caso Hotel Advisor

### Caso “Hotel Advisor”

Debido a la mala experiencia que tuvo Elvis Scott, un importante inversionista de empresas tecnológicas de Silicon Valley, en su último viaje a Chile, cuando se enfermó al tomar desayuno en el buffet de un hotel en Santiago, es que decidió invertir en una empresa chilena que realizaba un software en PHP para hacer recomendaciones acerca de hoteles. Scott, quien no se mide en gastos, arrendó una planta libre y contrató a los mejores UX designers, product owners, arquitectos, devops, y developers de Santiago, tanto de frontend como de backend, refundando así la empresa con un nuevo nombre: “Hotel Tech”.

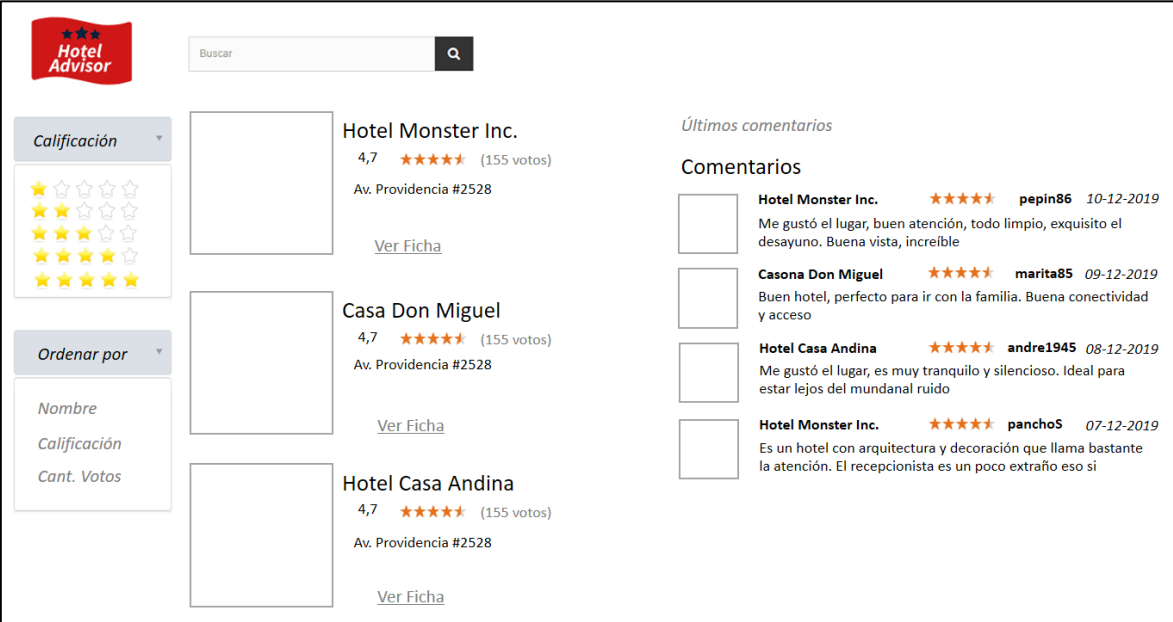
Scott ha quedado impresionado con la entrevista que le hizo a Usted y ha decidido contratarlo como Fullstack Developer Senior. Esto, debido a que Usted logró convencerlo que con Python y Django se puede implementar una aplicación Web escalable y robusta sin incurrir en excesivos tiempos de implementación.

### Requerimientos Funcionales del Sistema

Después de varias reuniones de trabajo con el Product Owner, los requerimientos que se tienen son los siguientes:

- Desarrollar una aplicación Web basada en el prototipo propuesto en las ilustraciones
- La pantalla de Home (**ruta: /**) debe permitir a los visitantes desplegar en el home, el listado de hoteles mostrando su imagen, nombre, calificación promedio, cantidad de estrellas, cantidad de votos y dirección. Nótese que la cantidad de estrellas muestra valores enteros entre 1 y 5 (es decir, si el promedio de calificaciones es 4.7, se muestran 5 estrellas) (*ver Ilustración 1: Listado de hoteles*).
- Opcionalmente debe poder mostrar los últimos comentarios ordenados por fecha.
- El sistema debe permitir realizar búsquedas de hoteles (por nombre), realizar filtros (por cantidad de estrellas), ordenamientos (nombre, promedio de calificación)
- En la pantalla del Hotel (**ruta: /hotel/<int:id>**) el sistema debe permitir ver la ficha de un hotel, en donde se ve la información general del hotel, sus atributos, y comentarios hechos por los demás usuarios
- El sistema debe permitir a los visitantes calificar al hotel con una nota de 1 a 5 agregando un comentario (**ruta: /reviews/<int:id>**). Cuando un usuario agregue una calificación, la aplicación posteriormente lo debe redirigir a la pantalla del Hotel.

- Cualquier usuario que visite la web podrá revisar las calificaciones y comentarios del hotel, sin embargo, para calificar o agregar un comentario deberá estar autenticado como usuario inscrito (pasajero). Esto sólo lo podrá hacer un usuario registrado previamente por el administrador del sitio.



The screenshot displays the Hotel Advisor interface. At the top left is the Hotel Advisor logo. A search bar with the placeholder 'Buscar' and a magnifying glass icon is located at the top center. On the left side, there are two dropdown menus: 'Calificación' (showing a star rating grid) and 'Ordenar por' (with options: Nombre, Calificación, Cant. Votos). The main content area lists three hotels, each with a placeholder image, name, rating (4.7), star rating (5 stars), number of votes (155), and address (Av. Providencia #2528). Each hotel entry has a 'Ver Ficha' link. On the right side, under the heading 'Últimos comentarios', there is a section titled 'Comentarios' showing three recent reviews with placeholder images, hotel names, star ratings, usernames, and dates.

Hotel	Rating	Stars	Votes	Address	Action
Hotel Monster Inc.	4,7	★★★★★	(155 votos)	Av. Providencia #2528	<a href="#">Ver Ficha</a>
Casa Don Miguel	4,7	★★★★★	(155 votos)	Av. Providencia #2528	<a href="#">Ver Ficha</a>
Hotel Casa Andina	4,7	★★★★★	(155 votos)	Av. Providencia #2528	<a href="#">Ver Ficha</a>

**Últimos comentarios**

**Comentarios**

Hotel	Rating	Stars	Username	Date	Comment
Hotel Monster Inc.	★★★★★	pepin86	10-12-2019	Me gustó el lugar, buen atención, todo limpio, exquisito el desayuno. Buena vista, increíble	
Casona Don Miguel	★★★★★	marita85	09-12-2019	Buen hotel, perfecto para ir con la familia. Buena conectividad y acceso	
Hotel Casa Andina	★★★★★	andre1945	08-12-2019	Me gustó el lugar, es muy tranquilo y silencioso. Ideal para estar lejos del mundanal ruido	
Hotel Monster Inc.	★★★★★	pancho5	07-12-2019	Es un hotel con arquitectura y decoración que llama bastante la atención. El recepcionista es un poco extraño eso si	

Ilustración 1: Listado de hoteles

Ruta: /





### Hotel Casa Andina

4,7 ★★★★★ (155 votos)

Av. Providencia #2528

Vive la mejor experiencia de Santiago en un hotel único y exclusivo del barrio Italia

- Estacionamiento: Sí
- Wi-Fi: Gratis
- Mucama: Sí
- Acepta Tarjetas de Crédito: Sí
- Acepta reservas: Con 24 horas
- Habitaciones para 2, 4, 6 y 8 personas
- Aire Acondicionado: Sí
- Traslado aeropuerto: Sí

Calificar

### Comentarios



**Maria** ★★★★★ 10-05-2019

Me gustó el lugar, buen atención, todo limpio, exquisito desayuno. Buena vista. Increíble



**Pipe86** ★★★★★ 21-06-2019






Buen Hotel, perfecto para ir con la familia. Buena conectividad y acceso

Ilustración 2: Ficha de Hotel

Ruta: /hotel/<:id>

### Hotel Casa Andina

Av. Providencia #2528



Calificar

Ilustración 3: Calificar e ingresar comentario

Ruta: /reviews/<:id>

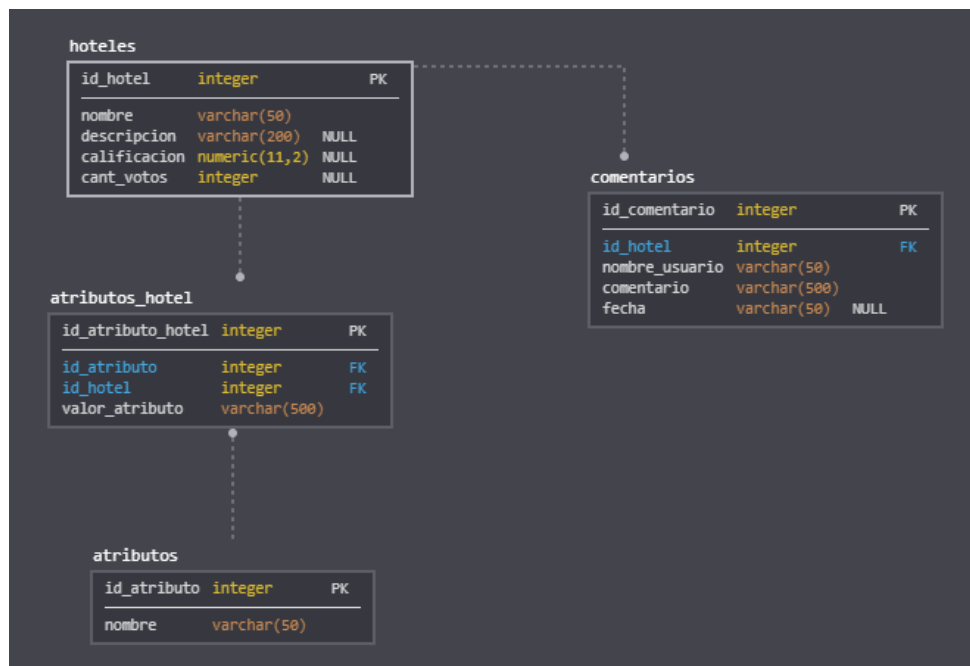
## Requerimientos No Funcionales

El arquitecto del proyecto, le ha hecho las siguientes definiciones:

- El sistema debe utilizar una base de datos PostgreSQL o MySQL.
- El sistema debe construirse con Python/Django
- El sistema Web debe construirse bajo el patrón MVC
- El acceso de usuarios se implementará con el modelo Login/Logout de Django
- La capa de acceso a datos debe ser implementada con el ORM Django

## Modelo de Datos

A continuación, se presenta el modelo de datos que ya ha sido diseñado por el equipo técnico del proyecto. Puede usarlo como guía para entender el problema, pero **no se preocupe** si los campos cambian de nombre al utilizar la Django ORM.



## Requerimientos a desarrollar

El líder de proyecto, en consenso con el arquitecto, le ha encargado que tome las siguientes responsabilidades:

## **1. Crear proyecto Django**

Cree un nuevo proyecto Django con todos los componentes requeridos para dar solución tanto a los requerimientos funcionales como no funcionales, las interfaces visuales, modelos y controladores. Lea los requerimientos no funcionales e incorpore todos los módulos que correspondan.

### **a. Crear proyecto Django**

Cree el proyecto Django con los comandos adecuados y la estructura necesaria para soportar los requerimientos del proyecto

### **b. Desarrollar vistas, modelos y controladores**

Desarrolle las vistas, modelos y controladores que sean necesarios para ajustarse a los requerimientos del proyecto. Base su diseño Web en el prototipo propuesto tratando de ajustarse lo más posible a la interfaz.

### **c. Desarrollar los formularios**

Elabore los formularios que necesite (puede utilizar la *FormClass* de Django). Cree las vistas/template de despliegue y los modelos que estime convenientes para cumplir el requerimiento

### **d. Conectar el proyecto a la Base de Datos SQL**

Conecte su proyecto con el ORM de Django a la base de datos SQL y genere los modelos que sean necesarios para ejecutar las migraciones que hagan el mapeo del modelo a la base de datos. Se solicita, además, que la migración incluya datos ficticios para llenar las tablas de hoteles, atributos, y atributos hotel, con el objeto que su aplicativo tenga datos maestros para las pruebas.

### **e. Desarrollar las Querys manuales**

Desarrolle las querys manuales en Django (SQL) que estime convenientes para resolver problemas de Joins entre tablas y desplegar los datos que necesite para la aplicación Web

## **2. Implementar Control de Acceso utilizando el modelo Login/Logout de Django**

Implemente la autenticación del sistema utilizando la aplicación preinstalada de Django "Auth". Desarrolle el módulo Login/Logout que viene incorporado en Django y acóplelo a la aplicación Web creada anteriormente.

Recuerde que el aplicativo tendrá una sección accesible por todo el público visitante y la acción de calificar hotel podrá ser realizada sólo por usuarios autenticados y previamente inscritos.

## **3. Implementar la administración de usuario utilizando el sitio administrativo de Django**

Si ya ha llegado hasta acá, implemente el sitio administrativo de Django utilizando la aplicación preinstalada “Admin”. Desarrolle el módulo y cree los usuarios necesarios para acceder a la zona privada de la aplicación Web (usuarios autorizados que pueden dejar valoración).

## Entregable

Haga todos los supuestos que estime conveniente para la resolución del caso y déjelos debidamente comentados en el código fuente en caso de requerirse. Recuerde, al final del proyecto entregar lo siguiente:

- Código fuente del proyecto
- El proyecto debe llamarse exactamente **hotel-advisor** (sin espacios ni mayúsculas).
- Export de la base de datos con esquema y datos, (le puede consultar al instructor a cargo). Este modelo debe llamarse exactamente **hotel-advisor.sql** (sin espacios ni mayúsculas), y debe ir dentro de la carpeta del proyecto.
- La contraseña del superuser de DjangoAdmin debe ser **admin** (en caso de no ser así, debe especificar el usuario y contraseña en un README.txt en la raíz del proyecto).
- Archivo requirements.txt con las dependencias del ambiente virtual.