

Semana 8

**Ingeniería de Software (PRY3211)**

Formato de respuesta

|  |  |
| --- | --- |
| **Nombre estudiante:** | |
| **Asignatura:** | **Carrera:** |
| **Profesor:** | **Fecha:** |

## 1. Introducción

El presente documento corresponde al Manual de Pruebas de Usuario del proyecto *Hotel Pacific Reef*, desarrollado durante la asignatura Ingeniería de Software (PRY3211).

En él se registran los procedimientos, resultados y observaciones de las pruebas funcionales realizadas al sistema.

Su objetivo es garantizar el correcto funcionamiento de cada módulo, validar los requerimientos implementados y confirmar que el producto final cumple con los criterios definidos en el *Product Backlog* y en el ERS.

## 2. Objetivo de las Pruebas

Verificar el correcto funcionamiento del sistema de reservas, clientes, habitaciones y login, asegurando que:

* Todas las funciones operen sin errores.
* Los datos se registren y consulten correctamente desde la base de datos SQL.
* Las validaciones de usuario y formularios sean efectivas.
* El sistema responda adecuadamente en el entorno local.

## 3. Alcance

Las pruebas se realizaron sobre la versión final del sistema **Hotel Pacific Reef**, desarrollada con Django (Python) y base de datos SQL local.

Se aplicaron pruebas funcionales de caja negra, enfocadas en la interacción del usuario con la interfaz del sistema.

No se incluyen pruebas de carga ni de estrés, ya que el sistema fue diseñado para entorno local.

## 4. Entorno de Pruebas

| **Elemento** | **Descripción** |
| --- | --- |
| **Entorno** | Localhost (servidor Django) |
| **Lenguaje / Framework** | Python 3.x – Django |
| **Base de Datos** | SQL local (SQLite) |
| **Navegador de pruebas** | Google Chrome / Microsoft Edge |
| **Sistema operativo** | macOS – Windows 11 (pruebas cruzadas) |
| **Control de versiones** | Git + GitHub |
| **Gestión ágil** | Trello |

## 5. Metodología Aplicada

Se utilizaron **pruebas funcionales de usuario** basadas en los requerimientos definidos en el ERS y las Historias de Usuario.

Cada caso de prueba se elaboró considerando:

* Entrada (input) esperada.
* Resultado esperado.
* Resultado obtenido.
* Estado final de la prueba (Exitosa / Fallida).

Todas las pruebas fueron ejecutadas por los integrantes del equipo y validadas de manera conjunta.

## **6. Casos de Prueba**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre Caso de Prueba** | Buscar y reservar habitación disponible | | | |
| **Nombre de Usuario Testing** | Juan | | Pérez | |
| **Correo** | juan@example.com | |  | |
| **Teléfono** | **xxxxxxx** | **xxxxxxxxx** |  | |
| **Epica 1**  Gestión de reservas | **Fecha**  05/10/2025 | **Hora**  17:45 |  | |
| **HU1** | Como usuario quiero buscar habitaciones disponibles según rango de fechas | | | |
| **HU2** | Como usuario quiero reservar una habitación disponible ingresando mis datos | | | |
| **HUn** | Ingresar nombre de HU que cubre la funcionalidad | | | |
| **Tipo de Usuario** | Cliente del hotel | | | |
| **Artefacto a probar** | Formulario de búsqueda y reserva (interfaz web Django) | | | |
| **Prueba a realizar** | Funcional y de experiencia de usuario | | | |
| **Tiempo de duración** | 3 minutos | | | |
| **Actividad a Realizar** | **Info. del paso a ejecutar** | | | **Registro de errores/Corrección** |
| **Paso 1** | Ingresar fechas de check-in y check-out | | | No se detectaron errores. Reserva almacenada correctamente en Oracle. |
| **Paso 2** | Hacer clic en “Consultar” | | |  |
| **Paso 3** | Seleccionar habitación “Doble” | | |  |
| **Paso 4** | Completar datos personales de Juan Pérez y confirmar reserva | | |  |
| **Paso n** | Ingresar instrucción del paso | | |  |
| **Observaciones** | | | |  |
| Reserva registrada correctamente, visible en /admin y en la base de datos. | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre Caso de Prueba** | Validación de fechas inválidas | | | |
| **Nombre de Usuario Testing** | Juan | | Pérez | |
| **Correo** | juan@example.com | |  | |
| **Teléfono** | **xxxxxxx** | **xxxxxxxxx** |  | |
| **Epica**  Gestión de reservas | **Fecha**  05/10/2025 | **Hora**  17:45 |  | |
| **HU3** | Como usuario quiero recibir un mensaje de error si ingreso una fecha de salida anterior a la de entrada. | | | |
| **HUn** | Ingresar nombre de HU que cubre la funcionalidad | | | |
| **HUn** | Ingresar nombre de HU que cubre la funcionalidad | | | |
| **Tipo de Usuario** | Cliente del hotel | | | |
| **Artefacto a probar** | Formulario de búsqueda y reserva (interfaz web Django) | | | |
| **Prueba a realizar** | Prueba funcional de validación de fechas. | | | |
| **Tiempo de duración** | 3 minutos | | | |
| **Actividad a Realizar** | **Info. del paso a ejecutar** | | | **Registro de errores/Corrección** |
| **Paso 1** | Ingresar fechas de check-in y check-out | | |  |
| **Paso 2** | Hacer clic en “Consultar” | | |  |
| **Paso 3** | Verificar mensaje de error | | | El sistema muestra el mensaje: 'La fecha de salida debe ser posterior a la de entrada.' |
| **Observaciones** |  | | |  |
|  | | | |  |
| Validación funcional y visual correcta. | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Nombre Caso de Prueba** | Visualización de tipo de cambio | | | |
| **Nombre de Usuario Testing** | Juan | | Pérez | |
| **Correo** | juan@example.com | |  | |
| **Teléfono** | **xxxxxxx** | **xxxxxxxxx** |  | |
| **Epica**  Gestión de reservas | **Fecha**  05/10/2025 | **Hora**  17:45 |  | |
| **HU4** | Como usuario quiero visualizar el valor actualizado del dólar dentro del sitio para referencia de precios | | | |
|  |  | | | |
| **HUn** | Ingresar nombre de HU que cubre la funcionalidad | | | |
| **Tipo de Usuario** | Cliente del hotel | | | |
| **Artefacto a probar** | Banner de tipo de cambio (API externa ExchangeRate). | | | |
| **Prueba a realizar** | Prueba funcional de integración y visualización. | | | |
| **Tiempo de duración** | 3 minutos | | | |
| **Actividad a Realizar** | **Info. del paso a ejecutar** | | | **Registro de errores/Corrección** |
| **Paso 1** | Ingresar al Home. | | | No se detectaron errores. |
| **Paso 2** | Verificar que aparezca '1 USD = XXXX CLP'. | | | Tipo de cambio visible en todas las vistas HTML |
| **Paso 3** | Confirmar actualización según API externa | | | Sin errores. Tipo de cambio visible en todas las vistas HTML. |
| **Paso 4** |  | | |  |
| **Paso n** | Ingresar instrucción del paso | | |  |
| **Observaciones** | | | |  |
| Integración con API externa exitosa y coherente con valores reales. | | | | |

## 7. Resultados de las Pruebas

Todas las pruebas fueron ejecutadas correctamente.

El sistema respondió conforme a lo esperado en todos los casos de uso.

No se registraron errores funcionales ni de validación durante la ejecución de pruebas en entorno local.

Se adjuntaron evidencias (capturas de pantalla) en el repositorio GitHub y en el tablero Trello del proyecto:

* GitHub: <https://github.com/Chiripio/hotel_S4>
* Trello: *(Tablero final de Hotel Pacific Reef – S9)*

## 8. Conclusión Final

El sistema **Hotel Pacific Reef** cumple con los requerimientos establecidos en el ERS y validados mediante pruebas de usuario.

Todas las funciones principales (login, clientes, habitaciones y reservas) se ejecutan correctamente en el entorno local Django + SQL.

No se detectaron errores críticos ni bloqueos, por lo que el sistema queda aprobado para entrega final en la Evaluación Transversal.

## 9. Observaciones Adicionales

* Se recomienda incorporar en futuras versiones:
  + Pruebas automáticas con PyTest.
  + Módulo de notificaciones por correo electrónico.
  + Integración con pasarela de pagos en línea.
* Los resultados y evidencias están documentados en GitHub y Trello.

## 10. Validación de Entrega

*“Todas las pruebas se realizaron exitosamente en entorno local (Django + SQL). No se detectaron errores críticos. El sistema cumple con los criterios funcionales definidos y se valida para entrega final.”*





**Duoc UC**