

©RESERVA - PROYECTO DE DESARROLLO DE SOFTWARE 1617A

SECCIÓN 1

CARLO MAGURNO

RESUMEN. El presente documento es una definición general del sistema a desarrollar para el actual semestre de desarrollo de software, incluye la explicación de los requerimientos funcionales de los principales componentes del sistema. Adicionalmente se presentan algunos datos para el desarrollo de entregas futuras y finalmente se definen las reglas de la evaluación a ser utilizadas durante la corrección del proyecto.

1. INTRODUCCIÓN

Reserva es el nuevo sistema para toda la gestión de líneas aéreas, el cual suplantará a todos los sistemas actuales. El sistema no solo ofrece funcionalidades desde el punto de vista de la línea aérea (BackOffice) sino también desde el punto de vista del cliente (FrontOffice) ofreciendo a través de diferentes tecnologías una experiencia nueva de avanzada para el usuario en general.

2. REQUERIMIENTOS

A continuación se presenta una lista con los requerimientos iniciales del sistema *Reserva*. De existir alguna ambigüedad en los requerimientos, la misma se deberá mitigar con el cliente.

Es importante notar que la palabra gestión en el presente contexto se refiere al desarrollo completo de las operaciones CRUD, es decir Agregar, Consultar, Modificar y Eliminar, dependiendo del caso Activar e Inactivar y posiblemente otros. De hecho, esta es una ambigüedad que debe ser aclarada por caso de uso con el cliente.

BackOffice

1. Módulo de ingreso, seguridad y *landing page*. (1)
2. Módulo para la gestión de aviones y sus características. (2)
3. Módulo para la gestión de rutas comerciales. (3)
4. Módulo para la gestión de vuelos y sus características. (4)
5. Módulo para la venta de boletos y ejecución de *check-in*, incluye el equipaje. (5)

6. Módulo para la gestión de comida y bebida del vuelo. (6)
7. Módulo para la gestión de extravío de equipaje. (7)
8. Módulo para la gestión de automóviles por ciudad. (8)
9. Módulo para la gestión de hoteles por ciudad. (9)
10. Módulo para la gestión de restaurantes por ciudad. (10)
11. Módulo para la gestión de ofertas y paquetes. (11)
12. Módulo para la gestión de usuarios. (12)
13. Módulo para la gestión de roles. (13)

FrontOffice

1. Registro de usuario, autenticación y recuperación de clave. (14)
2. Módulo para la búsqueda y compra de boletos. (15)
3. Módulo para la gestión de reclamos (*frontoffice y backoffice*). (16)
4. Módulo para la gestión de planificación de vacaciones (itinerario). (17)
5. Módulo para la gestión del diario del viaje. (18)
6. Módulo para la gestión de reserva de automóviles. (19)
7. Módulo para la gestión de reserva de habitación de hotel. (20)
8. Módulo para la gestión de reserva de restaurante. (21)
9. Módulo para la gestión de reserva de cruceros. (22)
10. Módulo para la gestión de revisión (*review*) de restaurante u hotel. (23)
11. Módulo para la gestión de configuración del *frontoffice*. (24)

3. DESARROLLO

1. El sistema debe ser desarrollado para un entorno web.
2. Cada grupo se encargará del desarrollo de uno o más funciones o módulos del sistema.
3. La especificación de las funciones del sistema deberán ser entregadas en forma de un documento ERS completo.

4. Es tarea de cada grupo detallar los requerimientos en forma de casos de uso, diagrama de casos de uso, diagramas de actividades, requerimientos funcionales, no funcionales, etc.
5. El diseño de cada componente se realizará utilizando el diagrama de clases y diagrama de secuencia en formato UML y el diagrama ER para el esquema de bases de datos. Este último puede realizarse también utilizando notación UML.
6. El diseño del sistema se presentará en clase de forma unificada (un solo diseño por aplicación).
7. La aplicación será desarrollada utilizando el paradigma de programación orientada a objetos.
8. Se utilizarán los estándares de programación definidos por el grupo en general, seleccionados a partir de los discutidos en clase.
9. Se manejarán las herramientas de control de versiones y gestión de errores Bitbucket o Github (por definir).
10. Se desarrollarán pruebas unitarias para cada entrega.
11. Cada grupo será el responsable de la integración y consistencia de sus casos de uso con el resto del sistema.
12. La integración entre los sistemas se realizará a través de servicios web.
13. Para cada entrega, el sistema deberá estar plenamente integrado y funcionando, de lo contrario no será corregido.
14. Para cada entrega, el sistema no podrá ser ejecutado desde el entorno de desarrollo.
15. La selección del lenguaje y/o los frameworks a utilizar debe ser una decisión tomada en grupo.
16. El enunciado del proyecto puede cambiar entre las entregas.
17. La comunicación será manejada a través de *Slack*.
18. Se recomienda el uso de una metodología.

4. REGLAS Y EVALUACIÓN

1. La colaboración entre los grupos de trabajo es imperativa y necesaria para la realización de este proyecto.
2. Se evaluarán las cualidades del software junto con las técnicas y prácticas vistas en clase.
3. El diseño del sistema deberá ser realizado por todos los grupos en conjunto.

4. Cada grupo será evaluado por los casos de uso desarrollados.
5. La evaluación será en grupo pero también individual.
6. Todos los grupos serán responsables por la homogenización, consistencia e integración de los componentes del sistema.
7. Cada grupo deberá definir sin ambigüedades las tareas de cada uno de sus integrantes. Es recomendable que cada integrante trabaje en casos de uso distintos.
8. El proyecto deberá ser entregado antes de la fecha y hora límite, de lo contrario no será evaluado.
9. Todos los errores o incidencias encontrados en la aplicación deberán ser manejados.
10. Todos los integrantes de los grupos deberán intentar realizar la misma cantidad de trabajo.

5. LAS ENTREGAS

A continuación se presenta la lista de cualidades del software, artefactos o prácticas a revisar durante cada una de las entregas.

5.1. Primera Entrega.

Para la primera entrega se deberá realizar un prototipo del sistema sin funcionalidad. Esta primera entrega se basa en diseñar la interfaz de usuario del sistema, junto con la usabilidad y la navegación del mismo. Una recomendación es que se dediquen a programar (solo la interfaz) y a aprender bien el uso de las herramientas y frameworks a utilizar incluyendo el uso del controlador de versiones.

Junto con esta entrega de deberá entregar el documento ERS y el DAS, completos.

La asignación de los grupos son los valores entre los paréntesis al final de cada módulo.

5.2. Segunda Entrega.

1. Sistema totalmente integrado y funcional.
2. Robusto.
3. Correcto.
4. Completo.
5. Usabilidad (estándares web).
6. Programado utilizando correctamente el paradigma de POO.
7. Estándares de programación.

8. Documentación de código.
9. Manejo de errores.
10. Pruebas unitarias con cobertura hasta de un 60 % por módulo.
11. Control de versiones de código fuente.
12. ERS - Especificación de Requerimientos del Software.
13. DAS - Documento de Arquitectura del Software.
14. El proyecto deber ser compilado en modo *Release* y ejecutado sin utilizar el entorno de desarrollo.

5.3. Tercera Entrega.

1. Sistema totalmente integrado y funcional.
2. Robusto
3. Correcto
4. Completo
5. Usabilidad (estándares web)
6. Arquitectado utilizando n capas
7. Diseñado utilizando patrones de diseño
8. Programado OO
9. Generalización del código (*Genericity*)
10. Estándares de programación
11. Documentación de código
12. Manejo de errores
13. Registro de logs de Errores y Debug a archivo haciendo uso de alguna librería
14. Pruebas unitarias con cobertura hasta de un 100 %
15. Control de versiones
16. Sistema desplegado en 3 servidores (capa de interfaz, capa transaccional, capa de datos)
17. ERS - Especificación de Requerimientos del Software.
18. DAS - Documento de Arquitectura del Software