

Proyecto #1 - TECAir

Instituto Tecnológico de Costa Rica
Área Académica Ingeniería en Computadores
Bases de Datos (CE3101)
I Semestre 2022
Valor 20%



Objetivo general

- Desarrollar una aplicación que permita manejar la descripción del problema descrito abajo y los requerimientos de software.

Objetivos específicos

- Aplicar los conceptos del modelo conceptual y relacional.
- Crear una Base de Datos en Postgresql para que permita el almacenamiento de los datos.
- Crear un servicio API para que centralice la funcionalidad.
- Crear dos páginas Web para que exponga la funcionalidad al usuario.
- Crear una aplicación móvil utilizando SQL Lite como Base de datos empotrada.
- Usar herramientas como Angular, Bootstrap, HTML5, CSS, Entity Framework, y Reporting Services, Cristal Reports o similar para desarrollo de reportes y PDF's.
- Crear un documento de instalación.
- Crear un plan de proyecto que evidencie: la participación en el equipo de trabajo asignado de acuerdo a su rol y la ejecución de lineamientos para trabajo en equipos.

Descripción del problema

Su equipo ha sido designado con la responsabilidad del desarrollo de un sistema que permita gestionar la administración de la aerolínea **TECAir**.

La empresa TECAir es una empresa en el mercado de transporte aéreo de personas de bajo costo y surgió luego de una larga negociación de la mejor universidad pública de Costa Rica con las aerolíneas comerciales del área, su objetivo principal es proveer a los estudiantes la mejor alternativa para que logren realizar sus becas en el exterior.

Por tanto, TECAir se ve en la obligación de proveer a sus clientes las mismas facilidades que presenta una aerolínea comercial (aunque su mercado meta son los estudiantes **no** se restringe a ellos únicamente).

La definición de requerimientos se define en la sección **Requerimientos de Software**. Es importante mencionar: en caso de retraso de la entrega de la aplicación la empresa contratada deberá pagar la multa de \$20.000 mensuales por cada semana de atraso.

A continuación, se describen las vistas de la solución que debe ser desarrollada:

- **Vista Reservaciones:** Ésta es la plataforma que permitirá a los clientes realizar todas las funciones normales de una página web de una aerolínea, buscar vuelos, reservar vuelos, pagar sus vuelos, realizar el pre-chequeo.
- **Vista Aeropuerto:** Ésta es la plataforma que permitirá a los funcionarios de la aerolínea poder realizar la apertura y cierre de vuelos, confirmación de reservaciones, asignación de maletas a un usuario etc.
- **App Móvil:** corresponde a la plataforma que utilizarán los usuarios para realizar la reservación y compra de sus boletos.

Requerimientos del Software

- Vista Reservaciones.
 - ◆ **Gestión de Usuario:** el sistema debe permitir la creación de distintos usuarios del sistema. Esta funcionalidad será utilizada por cualquier cliente de la aerolínea donde se solicitará información de contacto (Nombre completo, teléfono, correo y si es estudiante en caso de serlo solicitará la información de la Universidad donde estudia y su carnet que se utilizara para almacenar millas en su programa de lealtad).
 - ◆ **Búsqueda de Vuelos:** los pasajeros tendrán la oportunidad de buscar sus vuelos ingresando su aeropuerto de origen y su aeropuerto de llegada.
 - ◆ **Reservación de vuelos:** Una vez seleccionado el vuelo se procederá con la reserva y se le habilitará la opción de cancelar con tarjeta de crédito.
 - ◆ **Despliegue de promociones:** existirá un área dentro del sitio exclusivo para anunciar las promociones de la aerolínea
- Vista Aeropuerto.
 - ◆ **Gestión de Usuario:** Igual que en la vista reservaciones.
 - ◆ **Búsqueda de vuelos:** Igual que en la vista reservaciones.
 - ◆ **Reservación de vuelos:** Igual que en la vista reservaciones.
 - ◆ **Gestión de promociones (extra):** El sistema debe permitir ingresar promociones de un destino a otro, del cual se conoce Origen y Destino, precio en promoción y periodo de la promoción opcional una imagen de la promoción.
 - ◆ **Chequeo de pasajeros:** esta funcionalidad permitirá a los funcionarios del aeropuerto chequear un pasajero en un vuelo, debe ser flexible y poder seleccionar el lugar de preferencia del usuario en el **avión (asiento)** y finalmente **imprimir el pase de abordar** (donde se debe indicar la puerta de abordaje, hora de salida, asiento y número de vuelo) que puede: ser enviado por correo, enviado a una impresora o un dispositivo móvil.
 - ◆ **Asignación de maletas a un pasajero chequeado:** El sistema debe permitir crear una maleta y asignarla a un pasajero ya chequeado. De la maleta se conoce su dueño, el número de maleta que es único, el peso y el color.
El pasajero tiene derecho a una única maleta sin costo, por una segunda maleta se cobrarán 50\$ adicionales y después de la tercera maleta una tarifa fija de 75\$ adicionales.
Ej. Un Pax con 3 maletas tiene un cobro adicional de 125\$.
 - ◆ **Gestión de vuelos:** para poder registrar un vuelo se debe seleccionar **la ruta completa** desde el aeropuerto inicial hasta el aeropuerto destino y todos los aeropuertos intermedios que visite, así como la **matrícula del avión** que se utilizará para poder saber la cantidad de pasajeros que pueden reservar.

- ◆ **Apertura de vuelos:** esta funcionalidad permitirá a los usuarios del aeropuerto confirmar que los pasajeros efectivamente viajaran y con cuantas maletas para posteriormente realizar la carga y balance del vuelo.
- ◆ **Cierre de vuelos:** Una hora antes de la salida de cada vuelo se procederá a cerrar el vuelo esto con el fin de tener una lista oficial de los pasajeros finales que viajaran y cantidad de maletas del vuelo.

→ Aplicación Móvil.

- ◆ La aplicación móvil debe poder utilizarse sin conexión, por tanto se debe manejar una base de datos local en SQL Lite, que se sincronizará con la base de datos principal. Para el proceso de sincronización es indispensable contar con una conexión a la base datos principal.
- ◆ La funcionalidad de la aplicación móvil corresponde a la misma funcionalidad que la vista de reservaciones.

Aspectos adicionales:

- El término gestionar corresponde a las opciones de insertar, editar, eliminar y consultar.

Arquitectura del Software

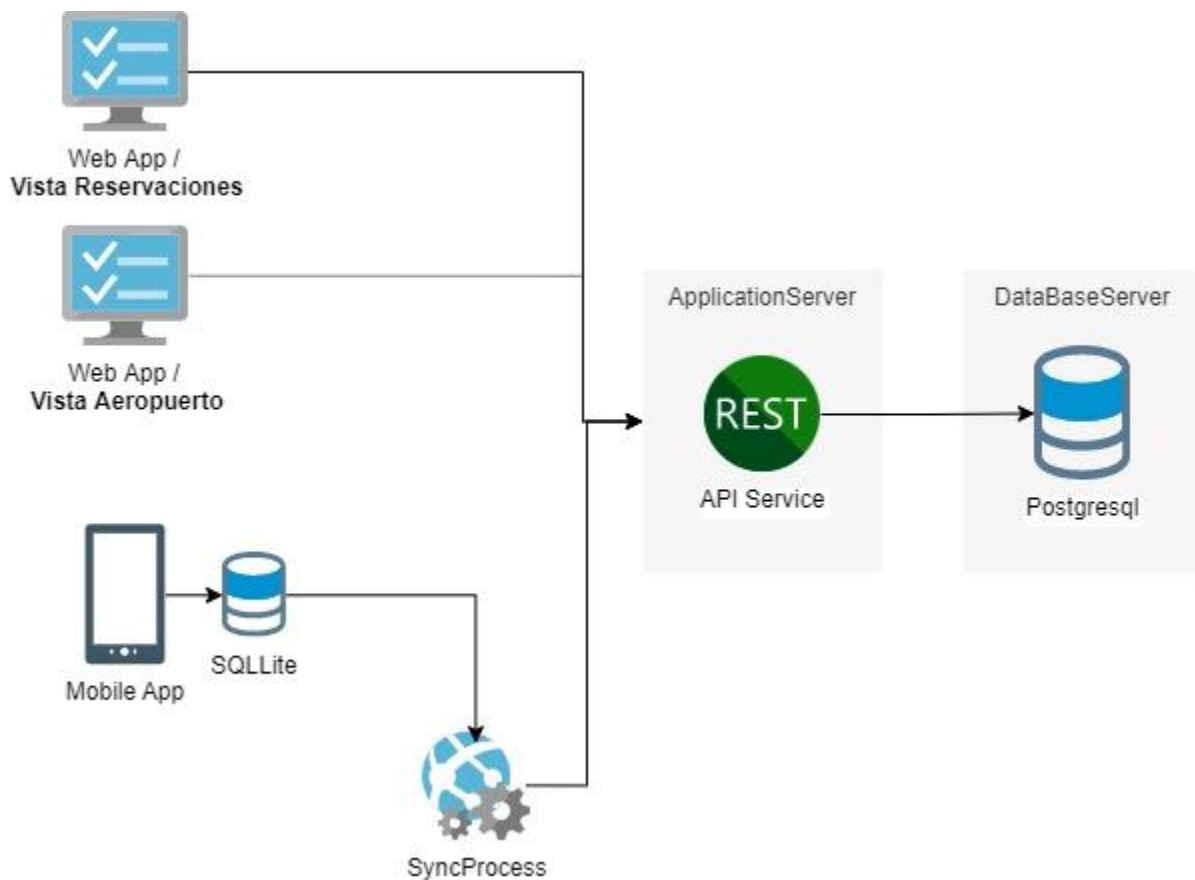


Ilustración 3 Arquitectura deseada.

Requerimientos no funcionales del sistema

- El Sistema debe ser una aplicación web (utilizando Angular, Bootstrap, HTML5, CSS y Entity Framework) y debe ser desplegada localmente en el **IIS**.
- La App Móvil debe ser desarrollada utilizando SQL Lite como base de datos.
- La Base de Datos debe estar en Postgresql.
- **No se** permite el uso de Procedimientos Almacenados, Vistas o Triggers. Los scripts de Base de Datos deben implementarse en la capa de datos (una librería en proyecto de C#).
- La capa de servicios debe estar desarrollada en C# y debe ser desplegada localmente en el **IIS**.
- El equipo de trabajo debe seleccionar a uno de sus miembros como único punto de contacto. Todas las comunicaciones y solicitudes deben ser a través de dicho punto de contacto.

Entregables

- Manual de Usuario.
- Documentación Técnica y del proyecto (especificada en el punto de **Documentación requerida**).
- Documento de instalación.
- Plan de Proyecto (Trabajo individual y en equipo).
 - ◆ Metas del proyecto: Describir los entregables asociados al proyecto.
 - ◆ Roles: Describir los roles y responsabilidades utilizados en el desarrollo del proyecto.
 - ◆ Reglas: Describir las reglas principales que han sido definidas para el trabajo en equipo.
 - ◆ Cronograma: Plan de proyecto con las actividades planeadas, responsables de cada actividad y fechas de entregas estimadas.
 - ◆ Minutas de las sesiones de trabajo: De manera que se evidencie la participación activa en el equipo de trabajo de acuerdo con el rol asignado y el seguimiento al plan de trabajo.
 - ◆ Bitácora: Evidencia la participación activa en el trabajo en equipo y la ejecución adecuada de los lineamientos para el trabajo en equipo, de cada uno de los integrantes del equipo. Deben describir las actividades realizadas como reuniones con el compañero de trabajo, investigaciones, consultas, entre otras. Se debe describir todo por más insignificante que sea. Esto demostrará el trabajo de cada uno de los miembros del equipo según el rol asignado.
- Script de creación de la Base de Datos.
- Script de población de Base de Datos.
- Aplicación WEB.
- Aplicación Móvil.
- Web API.

Documentación requerida

- Se deberá documentar el código fuente.
- Se deberá entregar un documento que contenga:
 - ◆ Modelo conceptual utilizando la notación de Chen.
 - ◆ Modelo relacional y la **descripción detallada del mapeo** de modelo conceptual al modelo relacional.
 - ◆ Descripción de las estructuras de datos desarrolladas (Relaciones).
 - ◆ Descripción detallada de la arquitectura desarrollada.

- ◆ Problemas conocidos: En esta sección se detalla cualquier problema que no se ha podido solucionar en el trabajo.
- ◆ Problemas encontrados: descripción detallada, intentos de solución sin éxito, soluciones encontradas con su descripción detallada, conclusiones, recomendaciones y bibliografía consultada para este problema específico.
- Conclusiones del proyecto.
- Recomendaciones del proyecto.
- Bibliografía consultada en todo el proyecto
- Diagrama de clases y un documento que explique el porqué del diseño.

Aspectos operativos y evaluación:

1. **Fecha de entrega:** De acuerdo con el cronograma del curso y lo establecido en el TEC Digital. Se establece el siguiente plan de entregas parciales:
 - a. Plan de proyecto: 25 de marzo 2022
 - b. Resumen Ejecutivo Avance 1: 1 de abril 2022
 - c. Resumen Ejecutivo Avance 2: 8 de abril 2022
 - d. Funcionalidad completa: 22 de abril 2022
2. El proyecto tiene un valor de 20% de la nota del curso.
3. El trabajo es **en grupos de cuatro personas**.
4. La implementación tendrá un valor de un 70% de la nota final, debe estar funcional. La defensa vale un 10% y la documentación un 20%.
5. Cumplir con los requerimientos especificados en la documentación no significa que se tienen todos los puntos, se evaluará que la documentación sea coherente, acorde al tamaño del proyecto y el trabajo realizado, no escatimen en documentación.
6. Cada grupo recibirá una nota en cada uno de los siguientes apartados Código y Documentación.
7. El profesor no sólo evaluará la funcionalidad del proyecto, esto quiere decir que aunque el proyecto este 100% funcional esto no implica una nota de un 100, ya que se evaluarán aspectos de calidad de código, aplicación del paradigma OOP, uso de herramientas solicitadas, calidad de documentación interna y externa, trabajo en equipo.
8. No se revisarán funcionalidades parciales, ni funcionalidades no integradas.
9. Es responsabilidad de cada miembro del grupo conocer su código, el profesor puede preguntar a cualquier miembro del grupo que le explique alguna funcionalidad/porción de código.
10. Las citas de revisión oficiales serán determinadas por el profesor durante las lecciones o mediante algún medio electrónico.
11. Aun cuando el código y la documentación tienen sus notas por separado, se aplican las siguientes restricciones
 - a. Si no se entrega documentación, automáticamente se obtiene una nota de 0.
 - b. Si no se utiliza un manejador de código se obtiene una nota de 0.
 - c. Si el código y la documentación no se entregan en la fecha indicada se obtiene una nota de 0.
 - d. Si el código no compila se obtendrá una nota de 0, por lo cual se recomienda realizar la defensa con un código funcional.
 - e. Si el grupo no cuenta con los equipos necesarios para realizar la revisión y no avisó al profesor de esta situación obtendrá una nota de 0.
 - f. El código debe ser desarrollado en C#, en caso contrario se obtendrá una nota de 0.

- g. Si algún estudiante no se presenta a la revisión se obtiene nota de 0.
12. Cada grupo tendrá como máximo 60 minutos para exponer su trabajo al profesor y realizar la defensa de éste, es responsabilidad de los estudiantes mostrar todo el trabajo realizado, por lo cual se recomienda tener todo listo antes de ingresar a la defensa.
 13. Cada excepción o error que salga durante la ejecución del proyecto y que se considere debió haber sido contemplada durante el desarrollo del proyecto, se castigará con 2 puntos de la nota final del proyecto.
 14. Cada grupo es responsable de llevar los equipos requeridos para la revisión.
 15. Durante la revisión únicamente podrán participar los miembros del grupo, asistentes, otros profesores y el coordinador del área.
 16. Las revisiones se realizan con los estudiantes matriculados en el curso, cualquier persona fuera de estos y los mencionados en el punto 15, no pueden participar en la revisión.
 17. Si no se realiza la defensa del proyecto se asignará una nota de 0 en el proyecto.

Referencias

AngularJS (2022-18-03). Recuperado de: <https://angular.io/>

Bootstrap Themes & Templates (2022-18-03). Recuperado de: <https://wrapbootstrap.com/>

How to Write Doc Comments for the Javadoc Tool. (2022-18-03). Recuperado de: <http://www.oracle.com/technetwork/articles/java/index-137868.html>

C# Coding Conventions (C# Programming Guide). (2022-18-03). Recuperado de: <https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/inside-a-program/coding-conventions>