

Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Estructura de Datos B
Ing. Carlos Alonzo
Aux. Bladimir Castellanos
Aux. Sebastian Oliva



Proyecto #1

Controlador de Vuelos y Aeropuertos

Objetivos

Crear una implementación de memoria dinámica a través de Objetos en Python y Java
Comprender y elaborar listas dinámicas y arboles AVL
Definir algoritmos de búsqueda y recorridos para las listas y arboles

Descripción

La industria de aeropuertos internacionales requiere un sistema en línea en el cual los usuarios puedan asignarse un vuelo o varios vuelos, también se requiere el sistema para el controlador de vuelos el cual tiene que notificar cuando el vuelo sale del aeropuerto, cuando se encuentra en vuelo y cuando el vuelo arriba al aeropuerto, mostrándoles a todos cuales vuelos están y donde están. Dado que la industria quiere saber el estado de los vuelos en ciertos tiempos se les dará un sistema local a las industrias para que realice los reportes de cada uno de ellos por medio de gráficas y pdf's. Uno de los problemas de la industria es que no tienen un servidor de base de datos y requieren que todo lo realicen en memoria dinámica.

Implementación

Se deberá crear una aplicación web con python, HTML que cumpla las descripciones para los sistemas de usuarios y controlador de vuelos, una aplicación de escritorio con java que cumpla los requerimientos de sistema local para la industria, y para la comunicación total un web services en cual contendrá todas las estructuras dinámicas, la parte visual queda a disposición del programador.

Sistema de usuarios

El sistema de usuarios debe realizarse en una aplicación web elaborada con HTML, python, los usuarios pueden realizar las siguientes acciones:

- Crear usuarios: El sistema debe soportar el ingreso de nuevos usuarios que incluyen los ingresos
 - Nombre

- Contraseña
 - Nombre de usuario (tiene que ser único y debe empezar por una letra seguido de cualquier alfanumérico)
 - Dirección
 - Teléfono
 - Número de tarjeta de crédito
 - Dirección actual
-
- Asignación de vuelos, los usuarios deben de poder seleccionar el vuelo que ellos desean (ver menú aeropuertos), e ingresar la categoría en la cual desean ir y si la categoría está llena no podrá asignarse, y si el vuelo ya salió del lugar donde se encuentra no se podrá asignar.
 - Observación de vuelos, los usuarios podrá observar todos los vuelos que hay en el sistema y el estado de los vuelos(ver menú aeropuertos)
 - Cambio de estado de dirección actual, este cambio lo debe realizar automáticamente el sistema
 - Se debe crear un listado de la información de los vuelos en el que se ha asignado el usuario, la información es: lugar donde salió, lugar donde llego, hora y fecha de llegada
 - El usuario se puede dar de baja, esto quiere decir que se eliminara por completo en el sistema y si quiere ingresar de nuevo tiene que crearse de nuevo

Sistema de controlador de vuelos

El sistema de controlador de vuelos debe realizarse en una aplicación web elaborada con HTML, python, los usuarios pueden realizar las siguientes acciones:

- Creación de aeropuertos, el sistema debe soportar que se puedan crear aeropuertos en el sistema los cuales necesitan los siguientes datos
 - Nombre del aeropuerto
 - País donde se encuentra
 - Contraseña mínimo 6 caracteres
- Creación de vuelos, cada aeropuerto debe crear los vuelos necesarios, los cuales llevan los siguientes datos
 - Lugar de llegada (El lugar de salida se sabe que es del mismo aeropuerto), y el aeropuerto debe existir sino este no puede ser asignado
 - Hora y fecha de salida
 - Hora y fecha de llegada
 - Cantidad y costo de pasajeros primera clase
 - Cantidad y costo de pasajeros clase turista
 - Cantidad y costo de pasajeros clase ejecutiva
 - Estado inicial el cual por defecto será "En aeropuerto"
 - El sistema debe crearle un id único

- Se debe tener un botón en el cual se diga que el vuelo correspondiente salió del aeropuerto, este no debe ser automático sino manual, y el estado del vuelo debe cambiar a “En vuelo”
- Se debe tener un botón en el cual se diga que el vuelo llegó al aeropuerto, este no debe ser automático sino manual, y el estado del vuelo debe cambiar a “En arribo”
- Eliminación del vuelo, el sistema debe proporcionar un botón en el cual se elimine el vuelo que este en los estados “En aeropuerto” y “En arribo” pero no en el estado “En vuelo”
- Modificación dado que los servicios de vuelos pueden tener un atraso estos se pueden modificar, a excepción de del id que el sistema se lo ingreso

Controlador de administradores

El controlador de administradores está elaborado en una aplicación java para escritorio, este se encargara de mostrar y reportar los estados actuales de los vuelos, y usuarios en cada uno de los aeropuertos, los reportes se darán de dos formas, la primera es la visibilidad en la aplicación y el segundo reporte es una exportación en PDF el cual se describirá lo mismo que en la pantalla visual. Así mismo en dicha aplicación se tendrá que realizar las graficar por medio de graphviz de las estructuras en el sistema, en este sistema no se tendrán las estructuras (ver web servicie) este deberá realizar las siguientes acciones:

- Muestra y reportar los usuarios y su información seleccionando uno de los vuelos que se encuentre en cualquier estado
- Muestra y reportar de los vuelos totales y toda su información que se encuentren en:
 - Vuelos “En aeropuerto”
 - Vuelos “En vuelo”
 - Vuelos “En arribo”
 - Todos los vuelos
- Muestra y reportar los vuelos que se encuentren en un cierto aeropuerto ingresando el nombre
- Muestra y reportar la información completa y vuelos donde se ha asignado un usuario
- Las gráficas de graphviz deben ser todas las estructuras ingresadas en el web services estas se deben crear en esta aplicación y descargar las estructuras.

Web Services

El sistema principal de toda la aplicación elaborada con web services la comunicación puede realizarse con SOAP, este se encargara de llevar el control de los vuelos, usuarios en memoria dinámica, utilizando las estructuras de árboles y listas, también debe controlar el contenido

enviado en cada una de los sistemas dado que se debe unificar todo y todos tengan el conocimiento de todo, las estructuras en cada una de las partes serán explicadas a continuación:

- Para los usuarios se debe crear un árbol AVL el cual está equilibrado por medio de los nombres de usuarios
- Para los vuelos se debe crear un árbol AVL el cual está equilibrado por medio de su id
- Para los usuarios asignados en un vuelo, se debe crear una lista doblemente enlazada adentro de cada nodo del árbol AVL de los vuelos esta lista solo se debe guardar el nombre del usuario
- Los aeropuertos se deben crear en una lista doblemente enlazada

Para la creación de los reportes de las estructuras que graphviz mostrara, se deben de realizar en base a las estructuras que se encuentran en el web service.

Observaciones

- Lenguaje a utilizar: Java con cualquier IDE y Python con Web Frameworks que mas les parezca (<https://wiki.python.org/moin/WebFrameworks>) y el web services se recomienda SOAP.
- Se calificará únicamente aplicación, no se realizará desde ningún IDE o herramienta de desarrollo.
- No se modificara código a la hora de calificación.
- Se debe de crear un repositorio en GitHub con el nombre de **Proyecto1s12015_#carnet los que no sigan dicha nomenclatura se le penalizara con 10% sobre su nota**
- Los documentos deben estar sincronizados don GitHub así mismo teniendo un historial de que trabajo en dicha plataforma
- Fecha y Hora de entrega: miércoles 17 de marzo del 2015 a las 10:00 am (Esta hora la última de sincronización en la plataforma) si esta esta sincronizada fuera de hora se penalizara con un 50% sobre la nota obtenida
- Forma de entrega: vía GitHub con notificación por correo de entrega (Asunto: EntregaProyecto y Cuerpo:#carnet) si no se notifica se penalizara con el 25% sobre la nota obtenida
- La herramienta para graficar será Graphviz (Solamente esta herramienta).
- Fecha de calificación pendiente
- El tiempo máximo para la calificación de dicha práctica será de 20 minutos.
- Para derecho de calificación se debe de poder ejecutar el programa y graficar todas las estructuras si no se grafican las estructuras se penalizara con un 50% sobre la nota
- Las estructuras deben de realizarlas por el estudiante, si se usa una estructura ya creada se anulara el proyecto y se colocara una nota de 0
- COPIAS SERAN PENALIZADAS con una nota de 0.
- COPIAS SERAN PENALIZADAS conforme al reglamento.