

PROYECTO PROGRAMACION ORIENTADA A OBJETOS

PRIMER PARCIAL

INTEGRANTES

- Gary Suarez
- Allison Larreategui

PROFESOR

- Ángel Lopez

PROYECTO

- Sistema de administración de aeropuertos.

Link del Repositorio: <https://github.com/gssuarez/ProyectoPoo>

DESCRIPCIÓN

El proyecto trata sobre el sistema de administración del nuevo aeropuerto de Guayaquil, el cual era usado por 3 tipos de usuarios por lo que se desarrolló un login general para todos los usuarios, para que luego fueran redirigidos a sus respectivos menús y desarrollen sus actividades.

FICHA TECNICA

- El password de todos los usuarios es encriptado mediante la librería Base64, para esto se desarrollo una clase con métodos estáticos de encriptación y desencriptación.
- La información se carga y guarda en archivos .txt para mantener la información luego de que el programa termine su ejecución.
- Los menús no son objetos instanciables, se usa sus respectivos métodos estáticos main junto a métodos estáticos para su mejor manejo.
- La clase sistema contiene toda la información del programa en ArrayList, el método cargarDatos lee todos los archivos para la funcionalidad del programa.
- Las clases y archivos se encuentran agrupados por tipos:
 - Archivos: contiene clase para leer archivos, archivos y clase para encriptar y desencriptar contraseña al momento de iniciar o finalizar el programa.
 - Entidades: contiene clases instanciables y las clases abstractas.
 - Menú: contiene todas las clases que ejecutan un menú.
 - Sistema: contiene la clase sistema que controla los datos del programa.

ABSTRACION

La abstracción se usa en el proyecto al momento de separar en métodos las diversas funcionalidades del sistema, por ejemplo cuando se usa el método *retornarPlanificacion(String código)* para encontrar la planificación mediante su código de vuelo.

ENCAPSULAMIENTO

El encapsulamiento se usa al momento de agrupar por paquetes las diversas clases, también cuando se usa correctamente los modificadores de acceso para no existan permisos innecesarios, ejemplo de esto podría ser la clase planificación a la cual únicamente se puede acceder a sus atributos mediante getters y setters.

POLIFORMISMO. HERENCIA E INTERFACES

Usamos herencia cuando creamos la clase abstracta Persona de la cual se genera la clase Usuario y cliente.