

Utilizamos Future de manera sencilla para concurrencia porque sabemos que nos facilita manejar las operaciones asíncronas. Nos permite ejecutar el cálculo de itinerarios de forma concurrente, aprovechando múltiples hilos de ejecución

Como vimos en clase, el método “par” sirve para convertir una colección en una colección paralela, permitiendo que las operaciones en esa colección se realicen de manera concurrente utilizando múltiples hilos, vimos que aplicaría para el cálculo de tiempos para múltiples itinerarios de vuelo entonces lo utilizamos y seleccionamos los tres mejores. Al este cálculo ser tan grande realizarlo en paralelo nos dio una mejora significativa en el rendimiento.

Pudimos evidenciar una mejora del rendimiento permitiendo aprovechar los recursos de hardware disponibles para ejecutar tareas concurrentemente, reduciendo así el tiempo total de ejecución. para convertir una colección en una colección paralela, permitiendo que las operaciones en esa colección se realicen de manera concurrente utilizando múltiples hilos, provee también eficiencia al optimizar el uso de recursos computacionales minimizando los tiempos de espera y maximizando la utilización de CPU durante el procesamiento de datos.

Con el benchmarking realizado se nota la diferencia y que tiene cierta escalabilidad pues a medida que es mayor la lista ha mejorado su rendimiento en comparación a la versión secuencial.

```
val (seq1, par1) = pruebas.itinerarios(itinerarioSeq.itinerarios(vuelosC1, aeropuertos), itinerarioPar.itinerariosPar(vuelosC1, aeropuertos))
val (seq2, par2) = pruebas.itinerarios(itinerarioSeq.itinerariosTiempo(vuelosC1, aeropuertos), itinerarioPar.itinerariosTiempoPar(vuelosC1, aeropuertos))
val (seq3, par3) = pruebas.itinerarios(itinerarioSeq.itinerariosEscalas(vuelosC1, aeropuertos), itinerarioPar.itinerariosEscalasPar(vuelosC1, aeropuertos))
val (seq4, par4) = pruebas.itinerarios(itinerarioSeq.itinerariosAire(vuelosC1, aeropuertos), itinerarioPar.itinerariosAirePar(vuelosC1, aeropuertos))
val (seq5, par5) = pruebas.itinerariosSalida(itinerarioSeq.itinerariosSalida(vuelosC1, aeropuertos), itinerarioPar.itinerariosSalidaPar(vuelosC1, aeropuertos))

println(saludo())
println(pruebas.mostrarPrueba(seq1, par1, "itinerarios"))
(args: Array[String])

ofc-proyecto-2024-lapp [:app.App.main()] x
:
541 ms > Task :app.App.main()
Unable to create a system terminal
Proyecto final
Tiempo de itinerarios: Secuencial: 0.8478 ms, Paralelo: 0.1225 ms, Aceleracion: 6.920816326530613
Tiempo de itinerariosTiempo: Secuencial: 24.4408 ms, Paralelo: 0.1007 ms, Aceleracion: 242.7090367428004
Tiempo de itinerariosEscalas: Secuencial: 3.7984 ms, Paralelo: 0.0288 ms, Aceleracion: 131.88888888888889
Tiempo de itinerariosAire: Secuencial: 2.0381 ms, Paralelo: 0.0391 ms, Aceleracion: 52.125319693094625
Tiempo de itinerariosSalida: Secuencial: 2.8776 ms, Paralelo: 0.049 ms, Aceleracion: 58.7265306122449
```

Nota: En cuanto a la prueba 4 para itinerarioSalida nos retorna una hora de salida mayor al que el test dice, tal como la lógica lo pedía y en cuanto a la hora de llegada llegan a la misma hora

Integrantes

Juan Miguel Palacios Doncel - 2359321

Nicolas Rodríguez Romero – 2266071

Jhon Alexis Ruiz Quiceno - 2266014