**CAPAS DE LA APLICACIÓN**

**Capas de la aplicación.**

El proyecto se realizó bajo arquitectura MVC.

Tiene un único punto de entrada que es el servicio REST (UploadFile.java), el cual a través de una petición POST que recibe un archivo bajo un parámetro llamado **file** y recibe un parámetro opcional llamado **trace** (que permite los valores true o false) para poder generar una salida detallada del proceso.

El servicio invoca el controlador (Challenge.java) quien se encargará de:

* Procesar el archivo recibido.
* Validar los datos de entrada, Shopping center, caminos y tipos de pescado.
* La estrategia de solución es primero buscar la ruta más corta para el Little Cat, y luego buscar la ruta que recorra los Shopping Center faltantes para completar la colección de tipos de pescado.

El modelo, define las características de cada uno de los elementos involucrados en la solución.

* Node.java es el Shopping Center
* StatusNode.java define el estado del nodo durante el proceso de búsqueda de la ruta más corta o requerida. Es decir, este componente está asociado al nodo y cambia cada vez que se evalúa la ruta.
* AdyacentNode.java guarda información referente al próximo nodo desde un nodo determinado. Es importante aclarar que cada nodo asocia múltiples nodos adyacentes.
* FishType.java almacena la información sobre el tipo de pescado y si ya se ha recolectado por alguno de los dos gatos.
* Result.java almacena la respuesta y su detalle, y es el objeto que devuelve el servicio REST.

**Clases**

|  |  |
| --- | --- |
| **Clase Paquete** | co.com.codingchallenge.main |
| App | Punto de entrada a la aplicación. |

|  |  |
| --- | --- |
| **Clase Paquete** | co.com.codingchallenge.rest |
| UploadFile | Servicio REST que permite la subida del archivo para resolverlo |

|  |  |
| --- | --- |
| **Clase Paquete** | co.com.codingchallenge.controller |
| Challenge | Controlador. Contiene de la lógica para resolver el problema de la recolección de los tipos de pescado dentro de un grafo no dirigido por parte de Little y Big Cat. |

|  |  |
| --- | --- |
| **Clase Paquete** | co.com.codingchallenge.model |
| Node | Modelo. Identifica los Shopping center. |
| StatusNode | Modelo. Identifica el estado del nodo durante la busqueda del recorrido para el Little y Big Cat. |
| AdyacentNode | Modelo. Identifica el tiempo (distancia) de recorrido entre un nodo y otro. |
| FishType | Modelo. Identifica un tipo de pescado. |
| Result | Modelo. Identifica el objeto que generará el servicio REST como respuesta. |

|  |  |
| --- | --- |
| **Clase Paquete** | co.com.codingchallenge.util |
| Util | Clase con métodos de utilería para manejo de archivos y tipos de datos. |
| RouteQueue | Cola de prioridad personalizada para permitir elementos irrepetibles. |

|  |  |
| --- | --- |
| **Clase Paquete** | co.com.codingchallenge.exception |
| ProjectException | Excepción personalizada para la aplicación. |