

POKÉDEX
registrarse(username, email, contraseña): void
iniciarSesión(user, pass): void
obtenerLista(tipo, usuarioLogueado): JSON
actualizarSeguimiento(usuarioObj): void
buscarUsuario(username): JSON
actualizarDatos(foto, email, contraseña): void
obtenerCuentasPendientes(): JSON
obtenerCuentasExistentes(): JSON

Controlador
getUserByTelegram(telegramID): string
getEquipo(userID): JSON
obtenerEstadísticas(): JSON5
obtenerRivales(string): JSON6
obtenerMejorPokemon(nombreEquipo): JSON3
obtenerCadenaEvolutiva(string): JSON6
obtenerPokemon(string): JSON3
vincularUsuario(userID, telegramID): void
obtenerEquiposNombreySprite(usuario): JSON1

GestorEquipo
getEquipo(userID): JSON
obtenerEquiposNombreySprite(usuario): JSON1
obtenerMejorPokemon(nombreEquipo): JSON3

GestorPokemon
obtenerPokemonEntero(string): JSON3
obtenerCadenaEvolutiva(string): JSON6
obtenerEstadísticas(): JSON5
obtenerRivales(string): JSON6
getEquipo(userID): JSON
getEquipo(userID): JSON
mostrarDetalles(JSONDetalles) : void

GestorListaPokemon
cargarListaPokemon(filtro) : void
almacenarListaCompleta(JSONLista) : void
filtrarListaCargada(criterio) : listaPokemon
mostrarLista(lista) : void
seleccionarPokemon(idPokemon) : void

SGBD
execSQL (sentencia: String): resultadoSQL
execSQL (sentencia: String): void

TelegramAPI
-token: string
-commando1(): void

NOTIFICACIÓN
-tipo: string
-texto: string
-fecha: Date

ESPECIE
-nombre: string
-ataque: int
-ataqueEsp: int
-def: int
-defEsp: int
-vida: int
-velocidad: int
-foto: image
-esLegendario: bool
-shiny: bool
-pokemons: ListaPokemons
-previo: Especie
-susAtaques: ListaAtaques
-tipos: ListaTipos
Especie(nombreEspecie, ataque, ataqueEsp, defensa, defensaEsp, velocidad, vida, listaTipos):void
estaPokemon(nombrePokemon):bool
addAtaque(nombreAtaque):void
addAtaqueAEspecie(nombreAtaque, nombreEspecie):void

GestorEspecies
-Lista: ListaEspecies
getImagen(nombrePokemon):image
existeEspecie(nombreEspecie):bool
addEspecie(nuevaEspecie):void
addAtaqueAEspecie(nombreEspecie, nombreAtaque):void
mostrarListaEspecies():void

AFFECTADO
-afectado: Tipo
-afecta: Tipo
-multiplo: float
Afectado(nombreTipo, otroTipo, multiplo)

GestorAfectados
-Lista: ListaAfectados
existeAfectado(nombreTipo, otroTipo, multiplo):bool
addTipo(nuevoTipo):void
addAfectado(nuevoAfectado):void

TIPO
-nombre: nombre
-susAtaques: ListaAtaques
Tipo(nombreTipo):void

GestorTipos
-Lista: ListaTipos
existeTipo(nombreTipo):bool
addTipo(nuevoTipo):void
addAtaqueATipo(nombreAtaque, nombreTipo):void

ATAQUE
-nombre: string
-damage: int
-descripción: string
Ataque(nombreAtaque,damage,descripcion,nombreTipo):void

GestorAtaques
-Lista: ListaAtaques
existeAtaque(nombreAtaque):bool
addAtaque(nuevoAtaque):void

EQUIPO
-nombre: string
-fechaCreación: Date
-pokemons: ListaPokemons
-número: int
Equipo(nombreEquipo)
obtenerPokemonNombreySprite(): JSON2
obtenerMejorPokemon(): JSON3
addPokemon(nuevoPokemon):void
addPokemon(nuevoPokemon,numRanura):void
mismoId(idequipo):bool
mostrarPokemons():void
eliminarPokemons():void
expulsarPokémon(numRanura):void

POKÉMON
-ID: int
-nombre: string
-ataque: int
-ataqueEsp: int
-def: int
-defEsp: int
-vel: int
-vida: int
-susAtaques: ListaAtaques
Pokemon(nombrePokemon, nombreEspecie, ataque, ataqueEsp, defensa, defensaEsp, velocidad, vida, idpokemon, listaAtaques)
obtenerMediaPokemon(): JSON4
obtenerNombreySprite () : JSON3
eliminar(idequipo):void
cambiarID(idPokemon):void
expulsar(idEquipo):void

USUARIO
-username: string
-email: string
-contraseña: string
-foto: image
-esAdmin: bool
-aprobado: bool
-cuentaTelegram: string
-notificaciones: ListaNotificaciones
-misEquipos: ListaEquipos
-seguidos: ListaUsuarios
-seguidores: ListaUsuarios
getNotificacionesSeguidos(): JSON
getNotificaciones(): JSON
setDatos(foto, email, contraseña): void
addEquipo(nuevoEquipo):void

GestorUsuario
- Lista: ListaUsuarios
getUserByTelegram(telegramID): string
getUserData(userID): JSON
vincularUsuario(userID, telegramID): void
obtenerNotificaciones(username: String): JSON
crearCuenta(username, email, contraseña): void
validarUsuario(user, pass): Usuario
obtenerLista(tipo, usuarioLogueado): JSON
actualizarSeguimiento(usuarioLogueado, usuarioObj)
buscarUsuario(username): JSON
actualizarDatos(foto, email, contraseña): void
obtenerCuentasPendientes(): JSON
borrarCuenta(usuarioSeleccionado): void
obtenerCuentasExistentes(): JSON
existeEquipo(username, idequipo): bool
cargarEquipo(username, idequipo):void
mostrarEquipo(username, idequipo):void
eliminarEquipo(username, idequipo):void
crearEquipoNuevo(username, nombreEquipo):void
addPokemonAEquipo(username, idEquipo, nombreEspecie, numRanura):void
expulsarPokemonDeEquipo(username, idEquipo, numRanura):void