



---

# PROYECTO DE SOC

---

Análisis de dos temporadas de la serie de Pokémon



13 DE ENERO DE 2023

GRUPO 13

Alejandro Valencia Blancae y Alonso Mata Fernández de Valderrama

Nuestro proyecto consiste en un análisis a la relación entre pokemons por aparición en 3 diferentes temporadas del anime; estas 2 temporadas son y pertenecen a la segunda temporada/Serie el comienzo, quinta temporada/Serie Oro y Plata. Estas han sido seleccionadas con separación entre ellas y buscando el mayor número de capítulos y pokemons posibles para así obtener redes ricas para analizar.

Obtención de datos: Para la obtención de los datos que hemos utilizado en este proyecto hemos utilizado como base una wiki (una página creada y modificada por fans) sobre pokemons, cuyo nombre es Wikidex, esta página es muy famosa entre la comunidad por lo que consta de una fiabilidad bastante elevada. La página que hemos seleccionado constaba de una lista de pokemons por aparición en la temporada “x”, esto nos facilitaba mucho el trabajo ya que haciendo uso de la librería pandas en python y sus funciones como puede ser: `pd.read_html('paginaalaquequierohacerscrapping.html')`. Haciendo uso de este comando hacemos scrapping de datos de la tabla html de la página, luego cogemos el data frame que nos interesa y lo exportamos en .csv separados por tabulaciones.

Más tarde haciendo unos pequeños ajustes a mano para que el formato del csv se quede igual en todas las líneas de las tablas (debido a pequeños hipervínculos en algunas palabras) hemos creado un rudimentario programa en c++ para que vaya leyendo esta primera base de datos y la convierta en un archivo nodelist.csv que contendrá la numeración, número de la pokédex, nombre, tipo1 y tipo2 de cada pokemon. Este programa (nodelistConverter.cpp) nos sirve para cualquier temporada de pokemon, ya que las diferentes páginas de esta wiki utilizan todos el mismo formato de tablas. Ahora tenemos que crear el archivo edgelist.csv el cual constará de tres dígitos separados por espacios, estos tres dígitos representarán la relación de dos pokemons y el valor de esta, con la forma Source, Target and Weight respectivamente, esto quiere decir que el primer dígito y el segundo muestran los identificadores de dos pokemons y el tercer dígito muestra el peso de esta relación, que en este caso es el número de capítulos en los que han aparecido juntos.

Para este segundo programa sólo necesitamos saber a priori el número de pokemons que aparecen esta temporada (para ahorrar memoria, se podría obviar al manejar números pequeños) y el número de pokemons “populares” que se refieren a que aparecen en todos los capítulos.

No me quiero explayar demasiado en la explicación del código como tal ya que es un programa muy rudimentario hecho específicamente para la extracción de datos de wikid pero al contrario que node.cpp este programa (edgelistConverter.cpp) no tan sólo lee y escribe. Lee los datos y los trata haciendo uso de estructuras de datos complejas para luego escribir el resultado que ansiamos.

Estas estructuras son un `map<string, map<int, string>>` diccionario; que es un mapa anidado en el que la key del primer mapa es el número de episodio y la key del segundo los pokémons y el valor de estos vacíos, así iterando sobre el valor que le corresponde a cada episodio (que sería el segundo mapa que contiene todos los pokemons que aparecen en ese episodio) tendría una forma de buscar por episodio que pokemons aparecen en este.

Para ir guardando cada relación entre los pokemons hago uso de una matriz que inicializo a 0 y que es del tamaño de los pokemons que aparecen esta temporada:

```
int matriz[POKEMONS][POKEMONS] = {0};
```

En esta matriz cada vez que el programa lee un pokemon y va leyendo los capítulos en los que aparece haciendo uso del mapa mencionado anteriormente va viendo si coincide con algún pokémon que ya haya leído antes y sí es así al tener guardado las ids tanto del pokemon que estamos leyendo como del pokemon que ya ha salido en este capítulo (esto último gracias al mapa) en la posición “matriz[pokemon1][pokemon2]++;” siendo pokemon1 el pokemon actual y pokemon2 el pokemon que ya sabemos que ha aparecido en el episodio. Siendo cada posición de la matriz la relación entre el pokemon1 y el pokemon2, si no han salido en ningún episodio juntos no existirá relación, si sólo han salido en un episodio en esa celda será 1 y si son más de 1 saldrá el número que requiera la relación.

Una vez tenemos la matriz completada ya tan sólo queda ir escribiendo en el archivo edgelist.csv y tendremos nuestro archivo de aristas perfecto.

Aquí le dejo la lista de episodios de la **segunda temporada** con los pokemons que aparecen en ellos:

```
KEY/EPISODIO: 052 Pokemons que aparecen en este episodio: 0 23 26 28 46 60 69 73 84 94 117 123 127 129 135 147  
KEY/EPISODIO: 053 Pokemons que aparecen en este episodio: 0 6 17 20 28 46 49 69 93 114 129 136 138 140 141  
KEY/EPISODIO: 058 Pokemons que aparecen en este episodio: 5 6 20 47 48 73 78 83 84 125 126 132 145 146  
KEY/EPISODIO: 059 Pokemons que aparecen en este episodio: 5 6 48 69 93 114 140 146  
KEY/EPISODIO: 060 Pokemons que aparecen en este episodio: 6 7 8 48 136 138 140 141  
KEY/EPISODIO: 061 Pokemons que aparecen en este episodio: 6 28 69 105 106 109 129 136 138 139 140 141 149  
KEY/EPISODIO: 062 Pokemons que aparecen en este episodio: 0 6 9 12 18 20 22 27 30 41 44 48 51 61 66 69 77 88 98 111 118 120 128 140 157 158  
KEY/EPISODIO: 063 Pokemons que aparecen en este episodio: 0 3 6 20 26 28 42 68 76 85 95 110 117 129 132  
KEY/EPISODIO: 064 Pokemons que aparecen en este episodio: 5 68 84 96 97 104 108 120 135 142  
KEY/EPISODIO: 065 Pokemons que aparecen en este episodio: 3 6 69 96 129 138 140 141 144 151  
KEY/EPISODIO: 066 Pokemons que aparecen en este episodio: 0 3 6 20 46 114  
KEY/EPISODIO: 067 Pokemons que aparecen en este episodio: 16 17 18 20 22 23 25 31 33 38 46 51 57 60 68 71 73 75 77 78 93 96 97 98 103 105 106 108 114 116 118 119 120 130 132 136 138 140 141 142 148 149 160 167  
KEY/EPISODIO: 068 Pokemons que aparecen en este episodio: 68 69 82 98 99 109 116 129 142 149  
KEY/EPISODIO: 069 Pokemons que aparecen en este episodio: 0 26 150  
KEY/EPISODIO: 070 Pokemons que aparecen en este episodio: 0 28 53 83 104 129 142  
KEY/EPISODIO: 071 Pokemons que aparecen en este episodio: 0 6 8 12 20 22 26 27 28 30 33 41 44 46 48 51 61 64 66 69 77 79 80 88 100 103 110 111 116 118 120 125 128 129 134 140 148 150 157 158  
KEY/EPISODIO: 072 Pokemons que aparecen en este episodio: 28 68 93 129 142  
KEY/EPISODIO: 073 Pokemons que aparecen en este episodio: 6 32 114 126 138 140  
KEY/EPISODIO: 074 Pokemons que aparecen en este episodio: 20 28 46 48 82 97 108 113 114 116  
KEY/EPISODIO: 075 Pokemons que aparecen en este episodio: 0 3 5 20 28 30 38 61 81 88 103 114 117 124 125 129 132 134 141 142 143 146 156 160  
KEY/EPISODIO: 076 Pokemons que aparecen en este episodio: 0 6 18 46 90 126 129 134 161  
KEY/EPISODIO: 077 Pokemons que aparecen en este episodio: 18 20 23 32 50 53 63 92 116 117 121 137 139 142  
KEY/EPISODIO: 078 Pokemons que aparecen en este episodio: 0 6 20 41 76 110 114 117  
KEY/EPISODIO: 079 Pokemons que aparecen en este episodio: 0 17 42 75 86 90 95 108 143  
KEY/EPISODIO: 080 Pokemons que aparecen en este episodio: 0 3 12 20 28 99 115 129 149  
KEY/EPISODIO: 081 Pokemons que aparecen en este episodio: 0 3 5 6 12 20 129 134  
KEY/EPISODIO: 082 Pokemons que aparecen en este episodio: 0 1 3 6 12 17 20 23 25 41 50 51 53 60 61 63 71 78 86 91 94 108 110 116 117 121 125 126 129 132 135 140 161  
KEY/EPISODIO: 083 Pokemons que aparecen en este episodio: 0 5 6 12 18 20 21 23 25 26 28 46 49 51 64 69 71 77 86 93 114 118 121 138 140 142  
KEY/EPISODIO: 084 Pokemons que aparecen en este episodio: 0 6 28 46 48 49 68 69 93 114 127 129 140  
KEY/EPISODIO: 085 Pokemons que aparecen en este episodio: 13 24 34 39 46 48 49 55 58 78 89 93 110 114 134 145 149 150  
KEY/EPISODIO: 086 Pokemons que aparecen en este episodio: 17 25 31 48 61 104 106 116 118 126 129 141 148 151  
KEY/EPISODIO: 087 Pokemons que aparecen en este episodio: 6 8 69 108 125 129 134 137 151  
KEY/EPISODIO: 088 Pokemons que aparecen en este episodio: 7 28 46 71 73 75 77 90 96 100 107 111 115 117 118 126 127 129 139 141 145 149 151 154 155 156  
KEY/EPISODIO: 089 Pokemons que aparecen en este episodio: 4 9 28 61 90 110 140 151 160  
KEY/EPISODIO: 090 Pokemons que aparecen en este episodio: 10 15 19 28 35 40 43 48 52 56 59 62 65 69 72 74 87 90 108 122 127 129 131 133 151 153 166  
KEY/EPISODIO: 091 Pokemons que aparecen en este episodio: 6 57 75 84 132 138 140 151 159  
KEY/EPISODIO: 092 Pokemons que aparecen en este episodio: 6 12 30 45 69 80 84 90 126 129 134 140 142 144 166  
KEY/EPISODIO: 093 Pokemons que aparecen en este episodio: 6 7 8 28 69 70 77 91 92 105 116 117 136 138 140 141 151 162 163 164 166  
KEY/EPISODIO: 094 Pokemons que aparecen en este episodio: 0 6 31 48 54 69 71 105 109 111 114 117 134 140 149 150 151 166  
KEY/EPISODIO: 095 Pokemons que aparecen en este episodio: 0 5 6 37 84 90 93 119 143 151  
KEY/EPISODIO: 096 Pokemons que aparecen en este episodio: 0 6 13 28 48 53 69 90 127 138 140 151 160  
KEY/EPISODIO: 097 Pokemons que aparecen en este episodio: 0 17 28 90 109 111 12 140 151 166  
KEY/EPISODIO: 098 Pokemons que aparecen en este episodio: 6 42 44 114 151  
KEY/EPISODIO: 099 Pokemons que aparecen en este episodio: 9 11 14 17 27 28 57 61 63 90 127 129 134 143 147 151 166  
KEY/EPISODIO: 100 Pokemons que aparecen en este episodio: 0 5 6 8 28 37 48 61 69 90 93 102 119 127 129 137 138 140 143 151 152 160 166  
KEY/EPISODIO: 101 Pokemons que aparecen en este episodio: 0 5 6 13 68 73 81 85 99 106 110 124 144 148  
KEY/EPISODIO: 102 Pokemons que aparecen en este episodio: 9 18 23 28 33 36 38 41 46 51 90 140 166  
KEY/EPISODIO: 103 Pokemons que aparecen en este episodio: 17 28 100 101 134 148 155  
KEY/EPISODIO: 104 Pokemons que aparecen en este episodio: 0 108 129  
KEY/EPISODIO: 105 Pokemons que aparecen en este episodio: 0 5 6 21 47 63 82 95 105 121 126 132 140 141 145 151  
KEY/EPISODIO: 106 Pokemons que aparecen en este episodio: 0 21 28 68 69 90 129 132 140 151 160 166  
KEY/EPISODIO: 107 Pokemons que aparecen en este episodio: 0 3 4 5 6 69 79 92 95 151  
KEY/EPISODIO: 108 Pokemons que aparecen en este episodio: 6 7 8 28 61 129 140 143 151 166  
KEY/EPISODIO: 109 Pokemons que aparecen en este episodio: 6 53 114 134 151 160  
KEY/EPISODIO: 110 Pokemons que aparecen en este episodio: 2 5 7 28 30 32 48 78 82 83 124 127 129 144 145 147 151  
KEY/EPISODIO: 111 Pokemons que aparecen en este episodio: 0 86 149 150 151  
KEY/EPISODIO: 112 Pokemons que aparecen en este episodio: 28 54 69 77 90 116 129 138 140 141 143 148 148 151 152 160 164  
KEY/EPISODIO: 113 Pokemons que aparecen en este episodio: 0 5 6 8 23 37 84 108 113 114 117 119 134 141 143 148 148 151 152 160 164  
KEY/EPISODIO: 114 Pokemons que aparecen en este episodio: 0 2 5 6 113 145 148 151 164  
KEY/EPISODIO: 115 Pokemons que aparecen en este episodio: 6 8 77 92 98 138 140 151 166  
KEY/EPISODIO: 116 Pokemons que aparecen en este episodio: 6 48 64 66 118 119
```

KEY/EPISODIO: 117 Pokemons que aparecen en este episodio: 13 36 37 41 46 48 49 54 90 93 102 104 108 114 117 127 130 142 148 160  
KEY/EPISODIO: 118 Pokemons que aparecen en este episodio: 12 17 42 60 61 63 76 81 85 114 130 132 142 143 145 148 150 153 166

## Quinta temporada:

KEY/EPISODIO: 212 Pokemons que aparecen en este episodio: 17 40 41 47 57 69 70 87 90 94 96 97 105 124 138 144 160 168 177 182 201  
KEY/EPISODIO: 213 Pokemons que aparecen en este episodio: 11 15 16 138 160 177  
KEY/EPISODIO: 214 Pokemons que aparecen en este episodio: 87 120 124 134 138 160 177  
KEY/EPISODIO: 215 Pokemons que aparecen en este episodio: 0 17 40 47 55 87 96 120 124 138 160 177 178 181 201  
KEY/EPISODIO: 217 Pokemons que aparecen en este episodio: 40 56 72 93 96 105 124 134 138 160 177 178 181 201  
KEY/EPISODIO: 218 Pokemons que aparecen en este episodio: 4 5 40 41 46 47 57 62 69 95 96 105 106 107 117 120 124 138 144 152 160 167 177 178 185  
KEY/EPISODIO: 219 Pokemons que aparecen en este episodio: 40 41 47 78 106 120 124 126 134 138 160 177  
KEY/EPISODIO: 220 Pokemons que aparecen en este episodio: 37 47 68 122 133 138 160  
KEY/EPISODIO: 221 Pokemons que aparecen en este episodio: 5 17 21 41 87 138 160 201  
KEY/EPISODIO: 222 Pokemons que aparecen en este episodio: 17 120 135 138 160 183 190 199 201  
KEY/EPISODIO: 223 Pokemons que aparecen en este episodio: 2 47 72 94 96 122 124 133 135 138 160 177 183 190 201  
KEY/EPISODIO: 224 Pokemons que aparecen en este episodio: 2 17 47 55 101 122 124 135 138 160 183 190 199 201  
KEY/EPISODIO: 225 Pokemons que aparecen en este episodio: 0 13 20 25 29 35 47 51 53 54 60 64 67 76 91 96 99 120 129 130 133 138 142 160  
KEY/EPISODIO: 226 Pokemons que aparecen en este episodio: 0 64 76 112 120 122 124 138 165  
KEY/EPISODIO: 227 Pokemons que aparecen en este episodio: 0 7 13 21 22 4 31 32 33 34 36 40 47 53 54 69 76 83 94 96 98 99 103 104 105 120 122 124 138 143 144 147 148 149 153 160 162 169 177 194  
KEY/EPISODIO: 228 Pokemons que aparecen en este episodio: 17 40 55 100 109 111 111 133 134 138 154 155 160 177  
KEY/EPISODIO: 229 Pokemons que aparecen en este episodio: 7 9 17 35 49 73 75 90 91 122 130 131 132 138 160 164 195 196 197 202  
KEY/EPISODIO: 230 Pokemons que aparecen en este episodio: 10 14 17 23 33 45 48 52 68 82 85 86 89 99 104 120 122 124 127 138 139 150 152 160 166 171  
KEY/EPISODIO: 231 Pokemons que aparecen en este episodio: 17 44 55 81 87 90 92 113 114 115 138 149 160  
KEY/EPISODIO: 232 Pokemons que aparecen en este episodio: 17 40 87 138 160 186  
KEY/EPISODIO: 233 Pokemons que aparecen en este episodio: 17 39 40 47 55 58 76 87 94 96 120 122 124 133 138 160 162 177 180 186  
KEY/EPISODIO: 234 Pokemons que aparecen en este episodio: 17 28 55 120 124 138 160  
KEY/EPISODIO: 235 Pokemons que aparecen en este episodio: 8 17 20 42 43 76 84 85 87 104 120 138 147 148 160 189  
KEY/EPISODIO: 236 Pokemons que aparecen en este episodio: 87 138 140 160 180  
KEY/EPISODIO: 237 Pokemons que aparecen en este episodio: 7 9 13 16 31 47 105 118 124 131 138 160  
KEY/EPISODIO: 238 Pokemons que aparecen en este episodio: 16 17 70 118 120 122 138 160  
KEY/EPISODIO: 239 Pokemons que aparecen en este episodio: 17 25 59 70 87 102 122 124 138 160 172 176 186  
KEY/EPISODIO: 240 Pokemons que aparecen en este episodio: 47 69 70 72 107 122 138 144 160 176  
KEY/EPISODIO: 241 Pokemons que aparecen en este episodio: 17 31 32 33 55 120 138 143 160  
KEY/EPISODIO: 242 Pokemons que aparecen en este episodio: 17 40 47 55 111 138 160 174  
KEY/EPISODIO: 243 Pokemons que aparecen en este episodio: 17 35 87 131 138 150 156 160  
KEY/EPISODIO: 244 Pokemons que aparecen en este episodio: 17 19 55 64 79 101 110 114 120 134 138 142 160  
KEY/EPISODIO: 245 Pokemons que aparecen en este episodio: 17 29 47 55 87 124 137 138 160 177 186  
KEY/EPISODIO: 246 Pokemons que aparecen en este episodio: 8 39 62 63 72 84 85 94 95 105 106 134 135 138 160 167 178 180  
KEY/EPISODIO: 247 Pokemons que aparecen en este episodio: 6 11 21 66 87 108 138 160  
KEY/EPISODIO: 248 Pokemons que aparecen en este episodio: 26 29 40 47 58 76 94 96 112 120 122 124 133 136 138 160 162 177 186  
KEY/EPISODIO: 249 Pokemons que aparecen en este episodio: 17 47 48 55 90 120 138 146 160 172 177 186 190  
KEY/EPISODIO: 250 Pokemons que aparecen en este episodio: 1 3 6 17 122 124 138 146 160 191  
KEY/EPISODIO: 251 Pokemons que aparecen en este episodio: 17 39 40 58 76 120 122 124 133 138 143 146 149 151 160 177 186  
KEY/EPISODIO: 252 Pokemons que aparecen en este episodio: 9 11 17 33 42 43 54 55 82 100 120 124 128 131 138 146 150 160 175 176 177 188  
KEY/EPISODIO: 253 Pokemons que aparecen en este episodio: 17 87 116 117 118 138 160  
KEY/EPISODIO: 254 Pokemons que aparecen en este episodio: 3 8 13 22 24 31 36 53 96 117 118 124 127 128 138 143 146 147 148 149 151 160 185 188 213  
KEY/EPISODIO: 255 Pokemons que aparecen en este episodio: 3 11 13 15 22 42 58 76 96 117 118 127 128 138 146 150 156 160 213  
KEY/EPISODIO: 256 Pokemons que aparecen en este episodio: 3 106 112 117 130 160 185 213  
KEY/EPISODIO: 257 Pokemons que aparecen en este episodio: 0 17 51 55 71 74 78 99 112 117 119 120 122 124 129 138 152 160 165 176 194 213  
KEY/EPISODIO: 258 Pokemons que aparecen en este episodio: 17 40 55 94 96 124 135 138 146 160 167 177 181  
KEY/EPISODIO: 259 Pokemons que aparecen en este episodio: 17 87 107 133 135 138 160  
KEY/EPISODIO: 260 Pokemons que aparecen en este episodio: 8 11 122 138 143 149 160 198  
KEY/EPISODIO: 261 Pokemons que aparecen en este episodio: 40 58 76 94 96 120 122 124 133 138 146 150 158 160 161 162 163 172 175 176 177 183 184 186 196 198  
KEY/EPISODIO: 262 Pokemons que aparecen en este episodio: 17 62 63 72 87 138 157 160 163 198  
KEY/EPISODIO: 263 Pokemons que aparecen en este episodio: 17 54 55 87 105 120 121 138 160 163 186 192 198  
KEY/EPISODIO: 264 Pokemons que aparecen en este episodio: 17 27 43 51 79 87 102 108 120 122 123 124 128 138 153 160 168 186 190 193 198  
KEY/EPISODIO: 265 Pokemons que aparecen en este episodio: 138 157 159 160 192 193 198 200  
KEY/EPISODIO: 266 Pokemons que aparecen en este episodio: 125 133 138 160 198 200  
KEY/EPISODIO: 267 Pokemons que aparecen en este episodio: 16 17 48 50 51 89 99 124 138 160 170 172 177 184 186 202 203  
KEY/EPISODIO: 268 Pokemons que aparecen en este episodio: 25 30 51 80 82 98 104 112 121 122 126 128 130 138 160 161 172 176 186 187  
KEY/EPISODIO: 269 Pokemons que aparecen en este episodio: 4 27 78 79 90 98 101 120 123 124 138 160 173 174 186  
KEY/EPISODIO: 270 Pokemons que aparecen en este episodio: 0 4 47 65 121 122 123 138 145 156 160 194  
KEY/EPISODIO: 271 Pokemons que aparecen en este episodio: 3 6 23 40 52 102 104 105 138 146 160 169 170  
KEY/EPISODIO: 272 Pokemons que aparecen en este episodio: 3 6 23 40 45 59 71 112 120 138 140 160 168 169 180 212  
KEY/EPISODIO: 273 Pokemons que aparecen en este episodio: 3 4 6 13 31 32 33 53 61 84 85 88 99 105 106 108 109 110 111 138 146 154 155 158 160 178 179 183 184 189 190 203 204 205 206 209 210 211  
KEY/EPISODIO: 274 Pokemons que aparecen en este episodio: 4 8 27 32 40 47 84 85 87 94 96 98 107 127 131 138 141 144 146 160 163 177 190 202  
KEY/EPISODIO: 276 Pokemons que aparecen en este episodio: 11 12 13 17 31 51 74 87 160 207 208 210 211

Podrá encontrar los archivos nodelistConverter.cpp y edgelistConverter.cpp dentro de la entrega del proyecto por si quiere revisarlo para comprenderlo por usted mismo.

Modelados de red: En nuestro modelo principal podemos decir que los nodos representan los pokemons y las aristas que los unen es la relación de que han aparecido en el mismo episodio.

Las aristas constan de pesos, este va desde 1 hasta 63 y desde 1 a 65 en la segunda y quinta temporada respectivamente, esto se debe a que en ambas temporadas hay dos o más pokemons que aparecen en todos los episodios de la temporada por lo que coinciden en todos, esto explica que el peso máximo de las aristas sea el número de capítulos de la temporada, lo que como podemos observar se cumple.

Nuestros grafos son no dirigidos, esto es fácil de entender al darnos cuenta que en la relación de que dos pokemons aparezcan en el mismo episodio no es direccional, con esto me refiero a que se puede recorrer de un nodo a otro de la misma forma.

El fenómeno que representan estas redes es un estudio sobre la relación de los pokemons dentro de la serie, la relevancia de estos y el porqué los guionistas decidieron darle tal importancia a estos pokemons. Básicamente es un estudio en profundidad sobre la serie, más en específico sobre la segunda y quinta temporada de esta, que son las temporadas con mayor número de capítulos. Buscamos crear conjeturas o hipótesis que puedan dar quizás respuesta a el porqué esta serie se hizo de esta forma y el porqué llegó a ser el fenómeno mundial en la que se convirtió.

Nuestras redes no se caracterizan por su cantidad de nodos precisamente, estas constan de 168 y 214 en la segunda y quinta temporada respectivamente. Esto no implica como veremos más adelante que tengamos un bajo número de enlaces ya que la gracia de nuestras redes, o lo más importante para analizar son las aristas que como podemos ver son muy numerosas, contando con 5196 y 5474 en la segunda y en la quinta respectivamente, esto nos da una idea de la gran densidad de las redes.

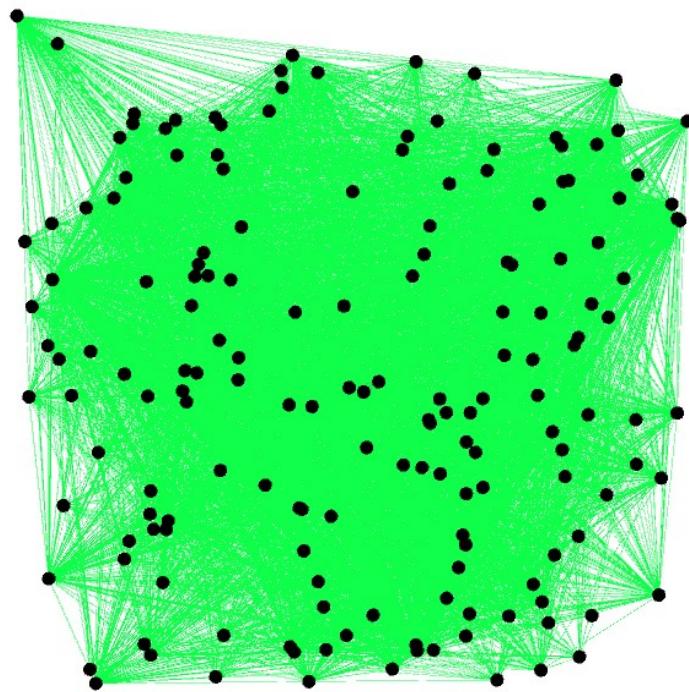
Cada nodo cuenta con varios atributos que nos van a servir para agrupar los nodos o para poder entender o analizar de mejor manera la red. Los nodos cuentan con ID, número de la pokédex, nombre del pokemon, tipo1 y tipo2. Principalmente el atributo más útil para nosotros van a ser los tipos ya que estos los podemos usar para agrupar a los pokemons e hipotetizar sobre la importancia de cada tipo de pokémon, cuál es el más numeroso o cual es menos y el porqué de estos descubrimientos.

Nosotros sólo vamos a modelar dos redes, la temporada 2 y 5 del anime, esto es una muestra bastante pequeña teniendo en cuenta que la serie actualmente cuenta con 25 temporadas de la línea principal y no tiene pinta de que vayan a dejar de producir nuevas temporadas a corto plazo, por lo que tenemos pokémons para rato. Sería interesante poder modelizar las redes de todas las temporadas y comparar datos o en su lugar crear una red enorme juntando todas las temporadas para poder hacer un análisis intensivo de qué pokémons a lo largo de estos 25 años han sido los más relevantes y que dirección ha cogido la serie últimamente, pero lógicamente para un equipo de 2 como somos no es factible realizar un trabajo tan ambicioso ya que tomaría mucho tiempo del cual no disponemos, pero es interesante contemplar la idea y saber que está ahí.

En este proyecto era muy deseable que las aristas tuvieran pesos ya que esto iba a indicar la fuerza de la relación entre dos nodos, refiriéndonos a fuerza con el número de capítulos en el que los dos pokémons han coincidido, esto nos da bastante más información que si tan sólo tuviéramos aristas planas de peso 1, ya que así podemos tomar en cuenta la relevancia de las relaciones.

A lo largo de todo el análisis sólo voy a explicar cada métrica una vez en la primera red que analice para evitar redundancia.

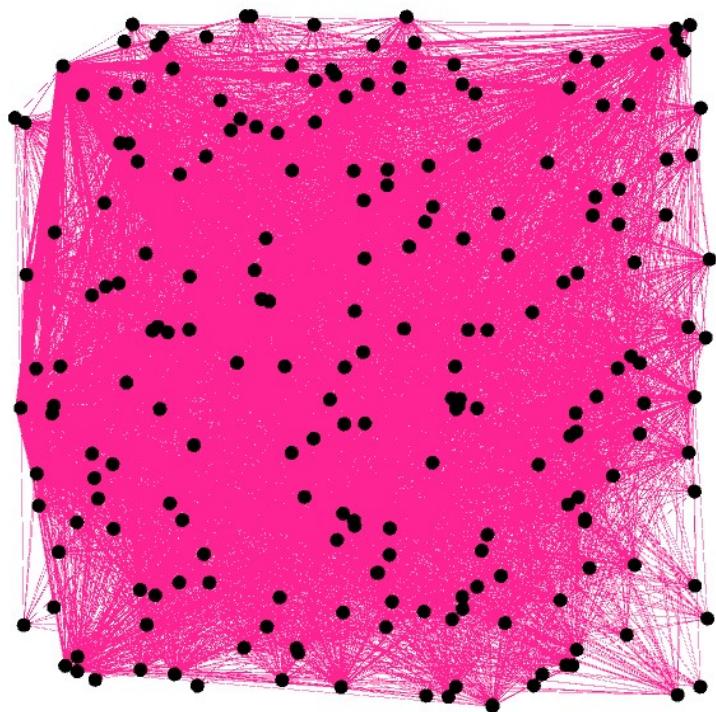
Análisis de la red:



Red 1 -> Temporada 2:

Nodos: 168

Aristas: 5196



Red 2 -> Temporada 5:

Nodos: 214

Aristas: 5474

Métricas Globales:

# Connected Components Report | T2

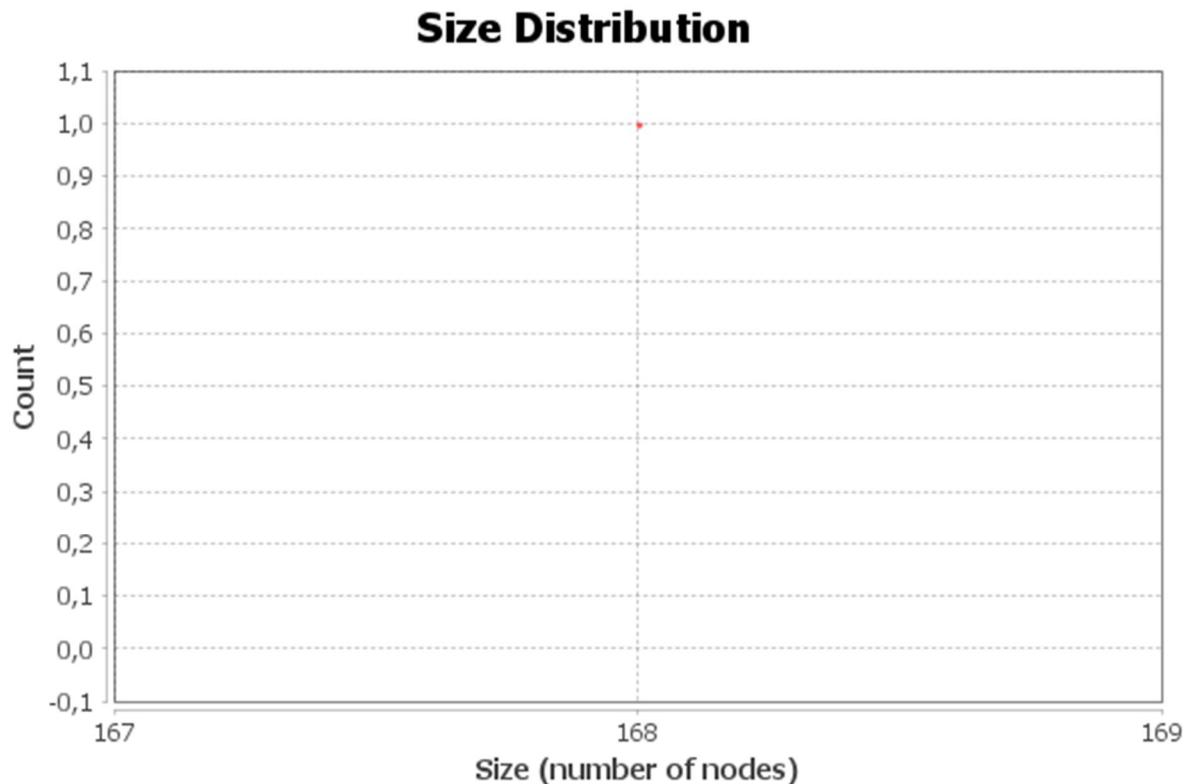
---

## Parameters:

Network Interpretation: undirected

## Results:

Number of Weakly Connected Components: 1



En teoría de grafos, un componente o componente conexo es un subgrafo inducido de un grafo en que cualesquiera dos vértices están conectados mediante un camino. Un vértice aislado, el grafo trivial o un grafo conexo son en sí mismos componentes.

En este caso es un grafo conexo, en teoría de grafos un grafo conexo o conectado es un grafo en que todos sus vértices están conectados por un camino, como es el caso por ello solo tiene una componente conexa, esto se debe a los pokemons que están conectados a todos los demás (principalmente los que aparecen en todos los episodios de la temporada).

## Algorithm:

Robert Tarjan, *Depth-First Search and Linear Graph Algorithms*, in SIAM Journal on Computing 1 (2): 146–160 (1972)

# Connected Components Report | T5

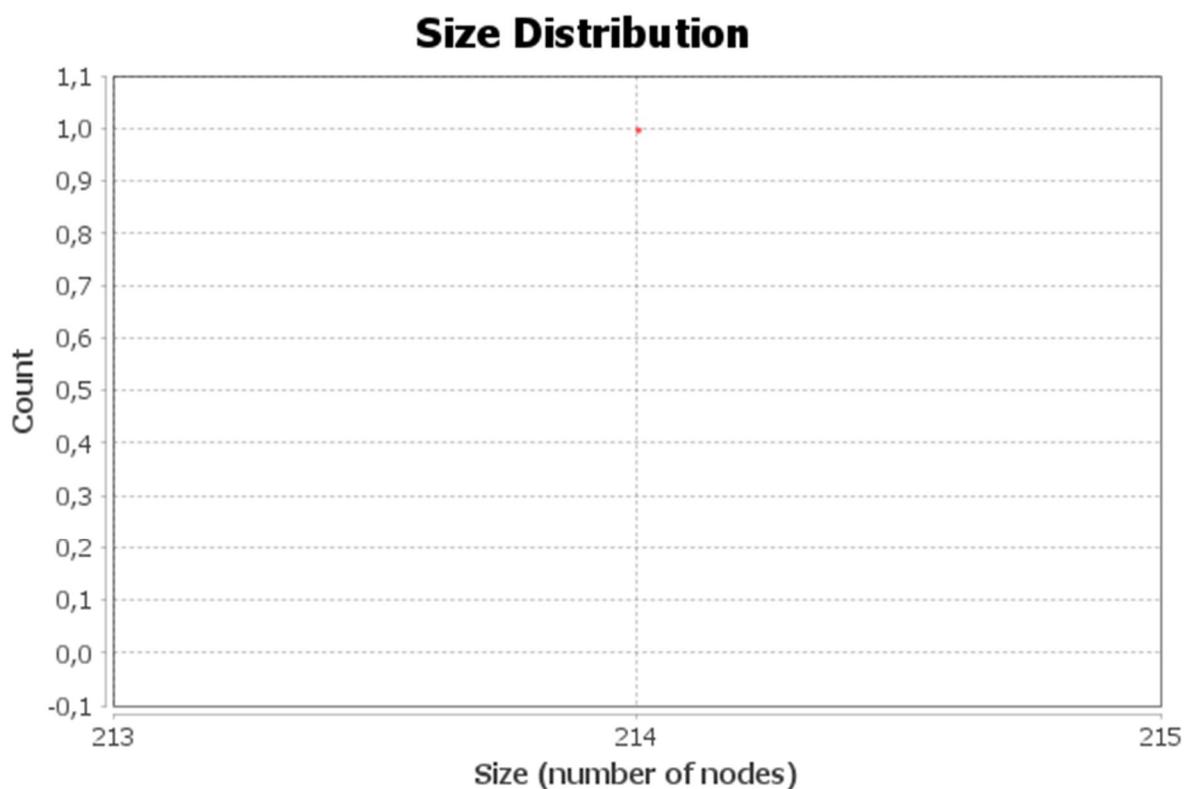
---

## Parameters:

Network Interpretation: undirected

## Results:

Number of Weakly Connected Components: 1



En este caso es un grafo conexo, en teoría de grafos un grafo conexo o conectado es un grafo en que todos sus vértices están conectados por un camino, como es el caso por ello solo tiene una componente conexa, esto se debe a los pokemons que están conectados a todos los demás (principalmente los que aparecen en todos los episodios de la temporada).

## Algorithm:

Robert Tarjan, *Depth-First Search and Linear Graph Algorithms*, in SIAM Journal on Computing 1 (2): 146–160 (1972)

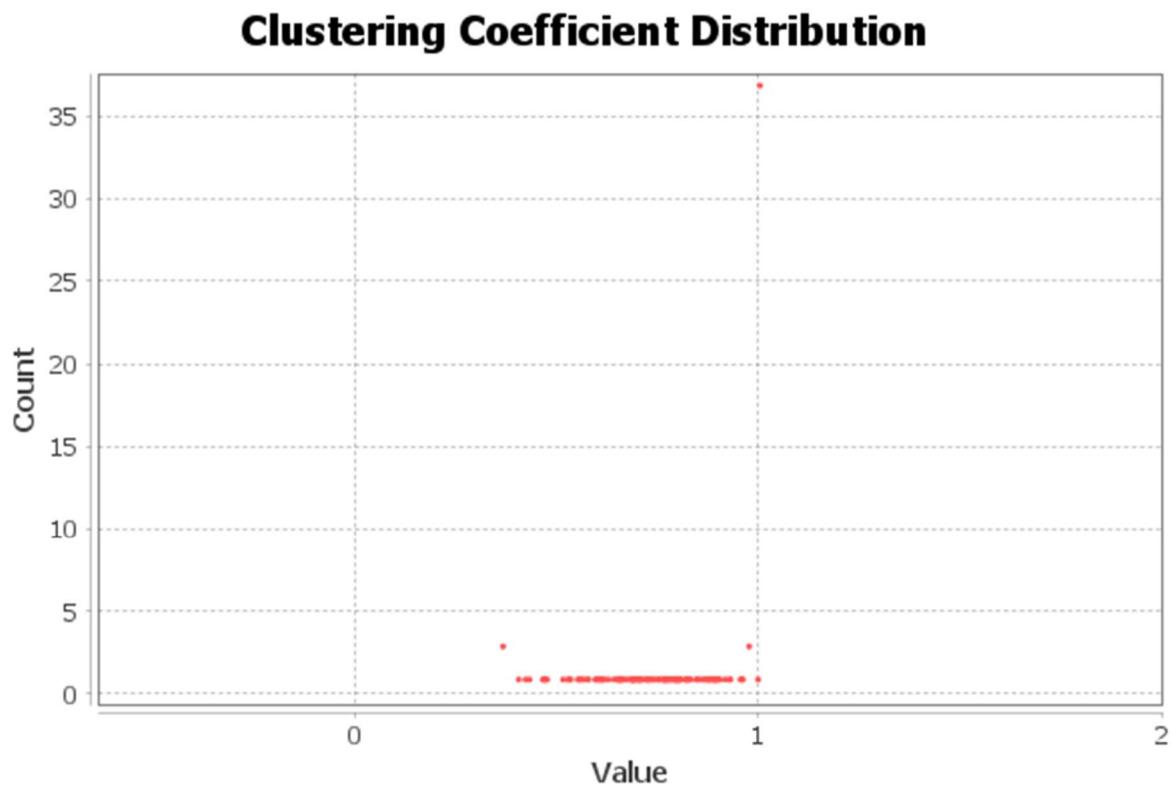
# Clustering Coefficient Metric Report | T2

## Results:

Average Clustering Coefficient: 0,784

Total triangles: 85890

The Average Clustering Coefficient is the mean value of individual coefficients.



# Clustering Coefficient Metric Report | T5

---

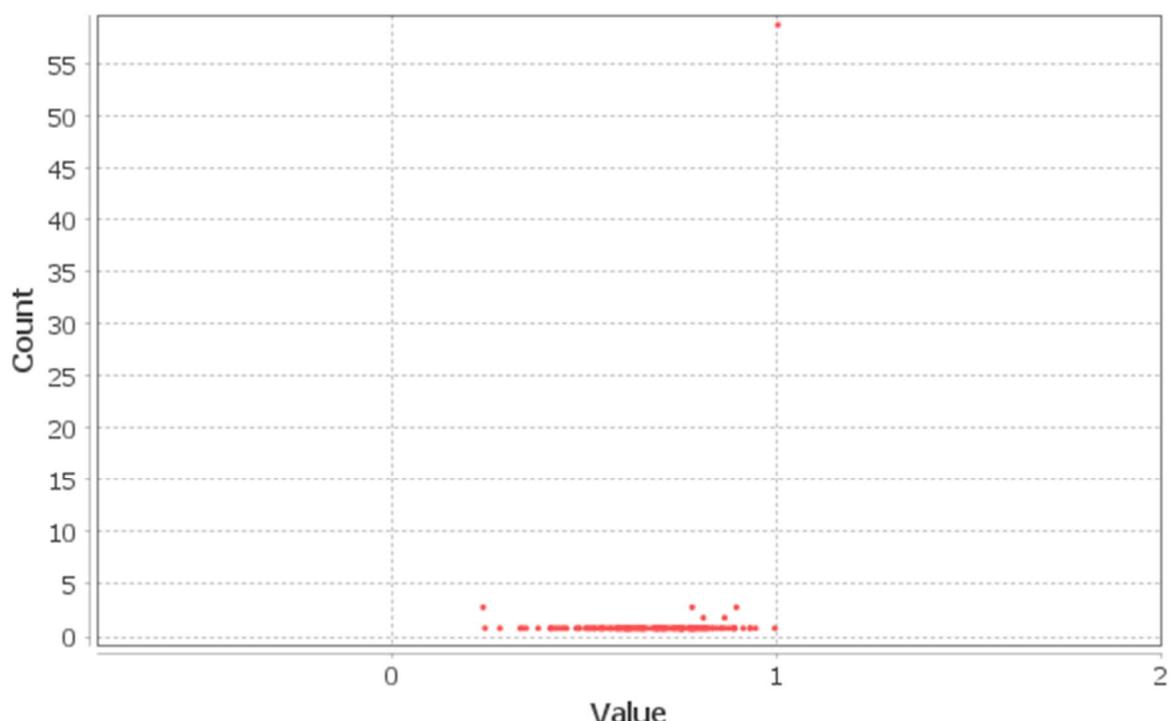
## Results:

Average Clustering Coefficient: 0,761

Total triangles: 70687

The Average Clustering Coefficient is the mean value of individual coefficients.

## Clustering Coefficient Distribution



En ciencia de redes, el coeficiente de agrupamiento (clustering coefficient, en inglés) de un vértice en un grafo cuantifica qué tanto está agrupado (o interconectado) con sus vecinos. Si el vértice está agrupado como un clique (subgrafo completo), entonces su valor es máximo, mientras que un valor pequeño indica un vértice poco agrupado en la red.

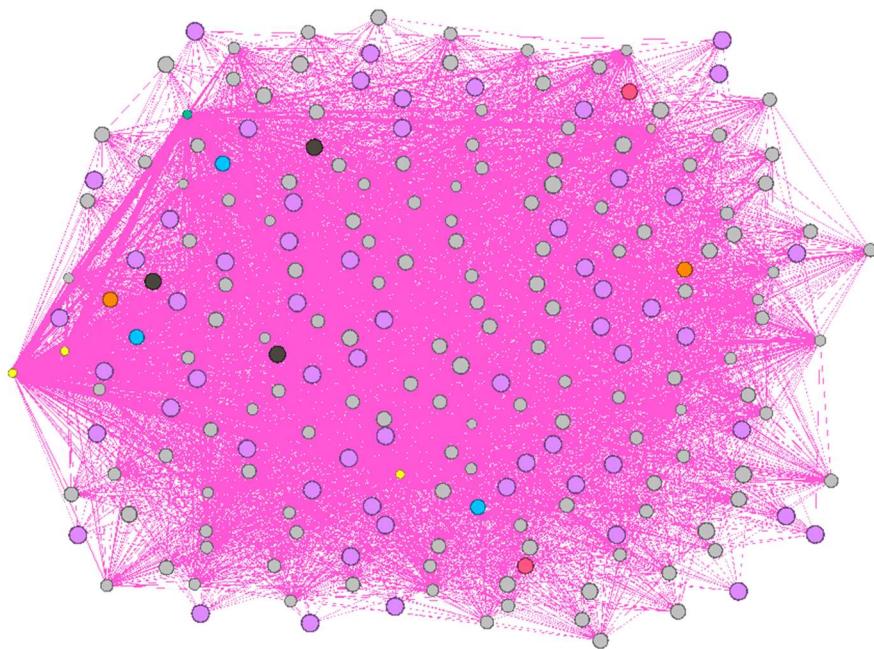
Al tener unos coeficientes de agrupamiento tan cercano a 1 (0,784 y 0,761 respectivamente) nos indica que los nodos vecinos suelen estar interconectados, relacionándolos con los pokemons esto quiere decir que los pokemons que aparecen en capítulos con sus compañeros y estos compañeros a su vez aparecen en capítulos con compañeros del primer pokemon. Se podría decir que hay “grupos de confianza” entre pokemons, queriendo referirme a que, si por ejemplo hay un pokemon “a” que ha aparecido en un episodio junto con un pokemon “b” y en otra ocasión ha aparecido en un episodio con otro pokemon “c”, hay muchas probabilidades que el pokemon b y c hayan aparecido en

algún episodio juntos. Esta idea es reforzada por el elevado número de triángulos que se encuentran en el grafo (85890 y 70687).

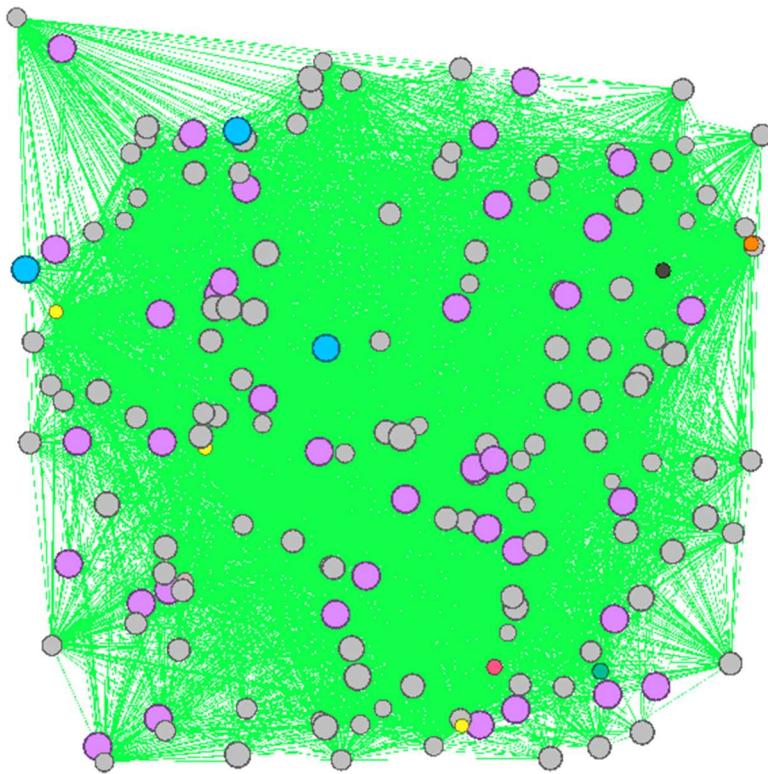
Aquí se puede ver de manera más gráfica:

Donde los nodos morados son los que tienen mayor coeficiente de agrupamiento (1) y en amarillo los que tienen el menor (0,2329)

T5:

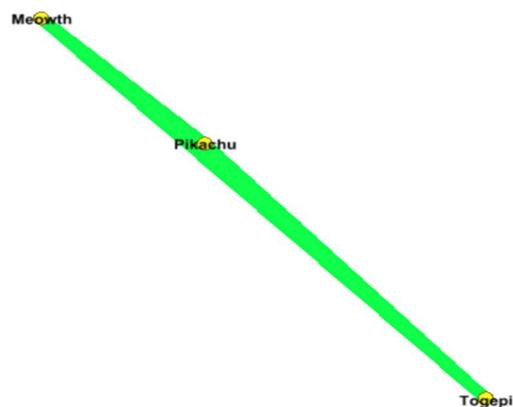


T2:



**Gephi: Partición clustering coefficient**

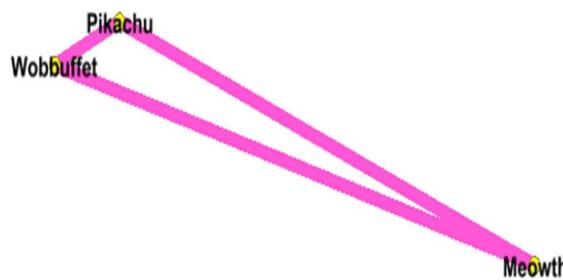
En ambas figuras podemos ver que los nodos con mayor coeficiente de agrupamiento (1) están pintados de morado y que los de menor clustering de amarillo (0.3627, 0.2329 respectivamente) podemos darnos cuenta de que los que menor coeficiente coinciden con los nodos con mayor grado esto no es una regla perfecta, pero se puede observar una tendencia clara.



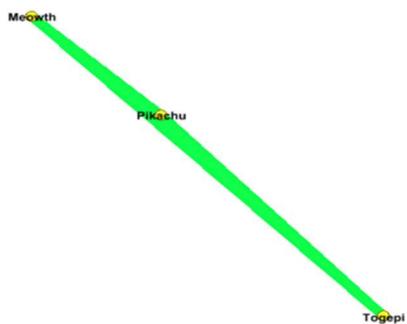
En la siguientes imágenes podemos darnos cuenta de lo anteriormente mencionado, tanto Pikachu como Meowth, Togepi y Wobbuffet tiene cardinalidad máxima (167 y 213 en nuestros casos ya que aparecen 168 y 214

pokemons en cada temporada); esto quiere decir que estos pokemons se relacionan con todos los demás pokemons que hay en la serie, osea que han aparecido en los suficientes episodios para coincidir con todos los demás.

T5:



T2:



#### Gephi: Partición y filtro clustering coefficient

Esto de que los que mayor cardinalidad sean los que menor clustering coefficient tiene sentido ya que cuantos más nodos vecinos más complicado se hace de que todos estos nodos vecinos estén conectados entre sí. Los que tienen “clustering coefficient” 1 al contrario son nodos normalmente con cardinalidad baja ya que cuantos menos nodos vecinos tenga un pokémon más fácil es que estos estén conectados entre sí, aunque es curioso, por ejemplo, el caso en la quinta temporada de “Cubone”, “Paras” o “Pinsir” que son los pokemons que comparten ser los pokemons con “clustering coefficient” 1 y mayor grado, este es 41. Al darme cuenta de que estos tres pokemons compartían el mismo número de triángulos, tal coincidencia me llamó la atención y fui a revisar la base de datos para ver qué ocurría y efectivamente, estos tres pokemons en toda la temporada sólo aparecen en un capítulo y coinciden los tres en este, es el episodio 227, por lo que virtualmente son idénticos excepto por los atributos de los nodos, pero van a tener aristas idénticas y la mayoría de las métricas iguales.

En la temporada dos por otra parte tenemos un claro ganador “Kakuna” que con grado 46 consigue tener un coeficiente de agrupamiento de 1, esto quiere decir que los 46 nodos con los que está conectado también están conectados entre sí, esto es bastante sorprendente teniendo en cuenta que es casi  $\frac{1}{3}$  del total de nodos de la red. Lo que nos da a

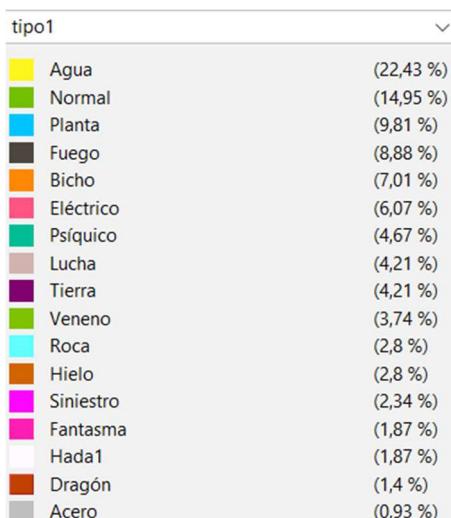
pensar como ya sabemos que esta segunda red tiene un clustering global mayor que la anterior,

También es curioso darnos cuenta que por regla general los nodos con mayor número de triángulos son los nodos con mayor grado como ya estábamos teorizando, son los que han salido en las figuras anteriores.

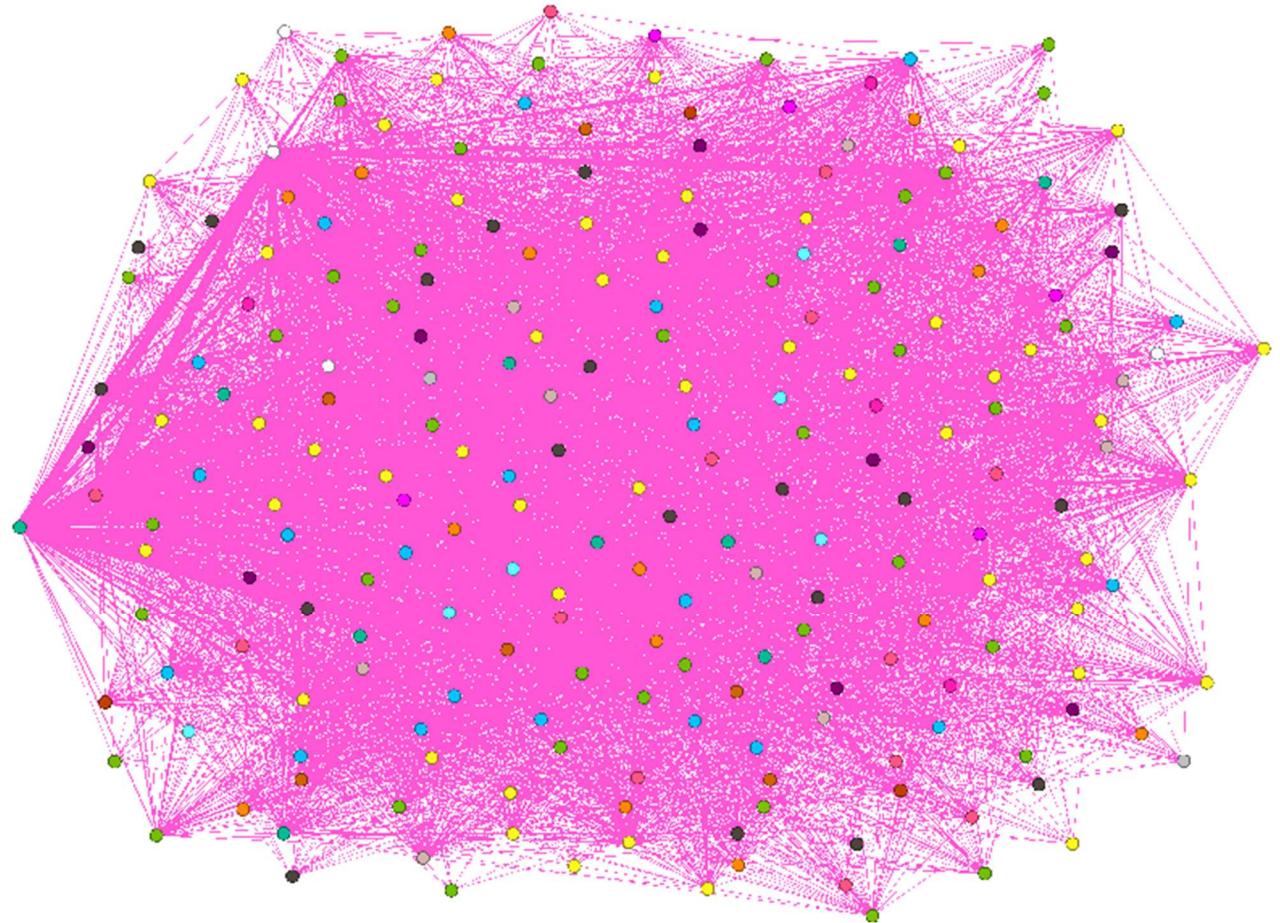
Los nodos tienen atributos que los caracterizan como son su número de la pokédex, nombre de pokémon, tipo 1 y tipo 2.

Aquí vamos a agruparlos por tipo 1, que cuenta con 17 posibles tipos de pokemons; aquí están los grafos de cada temporada que representan esta característica:

T5:



Gephi: Partición por atributo



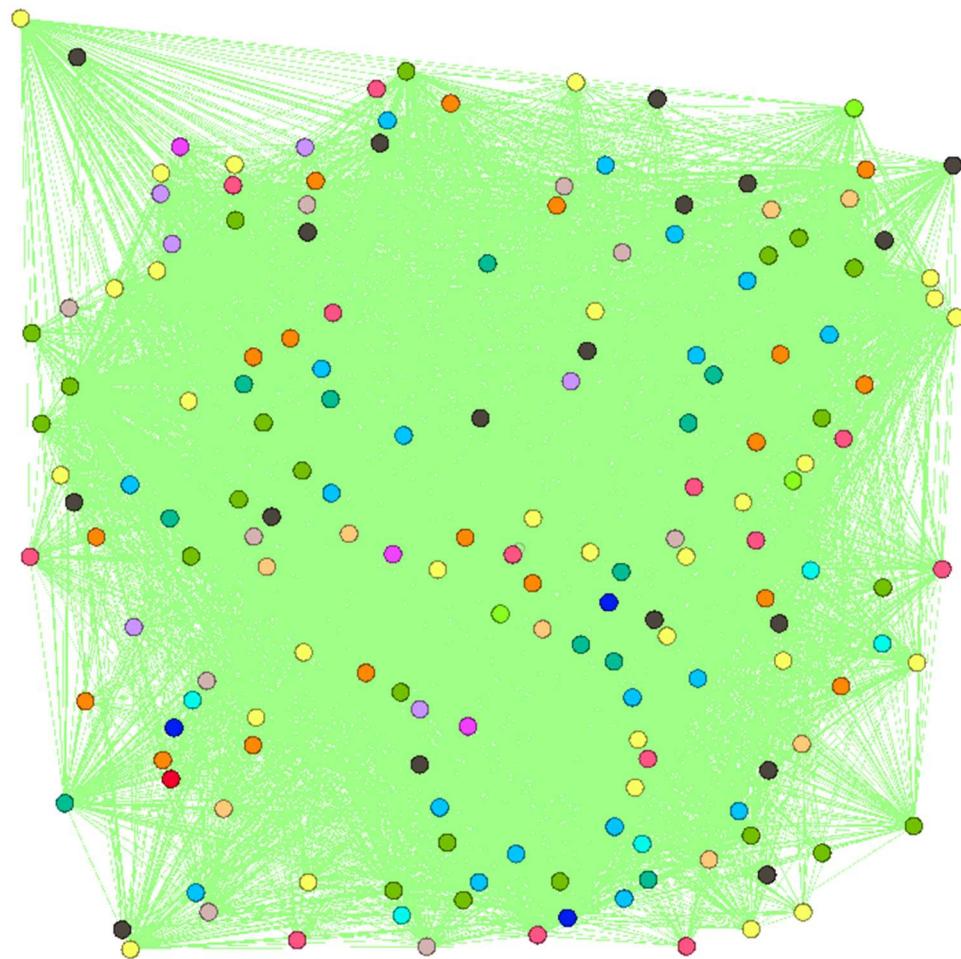
Como podemos observar en la figura anterior el tipo agua es el tipo predominante por diferencia contando con casi  $\frac{1}{4}$  del total de nodos en su grupo. Esto nos facilita saber por ejemplo, sabiendo que en pokémon hay unos tipos que son más fuertes contra otros tipos y a la inversa, escoger tipo fuego en esta temporada no sería una buena opción por la cantidad de enemigos de tipo agua que hay y que nos harían daño crítico.

Es interesante pensar en porqué se escogió este tipo como el predominante, ¿Por qué era el tipo más popular? ¿Por qué es el tipo más numeroso de pokemons? ¿Por qué es el tipo más efectivo? Todas estas son conjeturas, ya que es difícil estar en la mente de los guionistas o saber el porqué hicieron lo que hicieron.

T2:

tipo1	
Agua	(17,26 %)
Normal	(13,1 %)
Veneno	(11,31 %)
Bicho	(10,71 %)
Planta	(10,71 %)
Fuego	(7,74 %)
Tierra	(6,55 %)
Lucha	(5,36 %)
Eléctrico	(4,76 %)
Roca	(3,57 %)
Psíquico	(2,98 %)
Fantasma	(1,79 %)
Hada2	(1,79 %)
Dragón	(1,79 %)
Hielo	(0,6 %)

Gephi: Partición por atributo



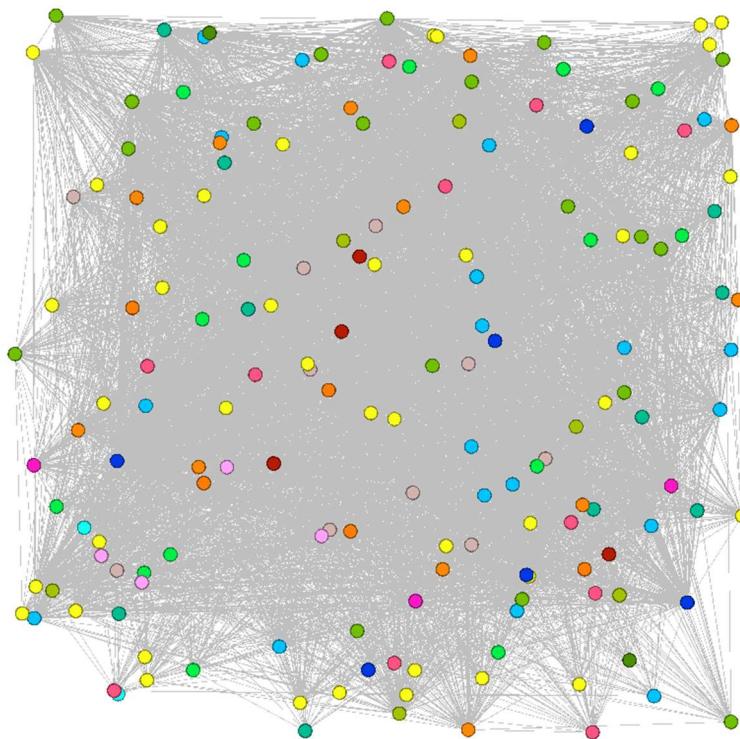
En esta temporada se repite que el tipo agua es el principal contando con un 17,26% un poco menos que en la temporada 5 podemos decir basandonos en estas dos temporadas que el tipo Agua es el predominante en la serie de pokémon seguido de cerca por el tipo normal y que en poco a poco la diferencia del tipo agua con el resto va aumentando, habiendo aumentado un 5,16% en 3 temporadas.

Los pokémons son de un tipo o de dos, ahora analizaremos el segundo tipo de estos:

T2:

tipo2	
Veneno	(23,81 %)
Aqua	(11,9 %)
Volador	(11,31 %)
Tierra	(8,33 %)
Normal	(7,14 %)
Psíquico	(6,55 %)
Fuego	(5,95 %)
Lucha	(5,95 %)
Eléctrico	(3,57 %)
Hada	(3,57 %)
Planta	(2,98 %)
Bicho	(2,38 %)
Roca	(2,38 %)
Hielo	(1,79 %)
Alero	(1,19 %)
Dragón	(1,19 %)

### Gephi: Partición por atributo



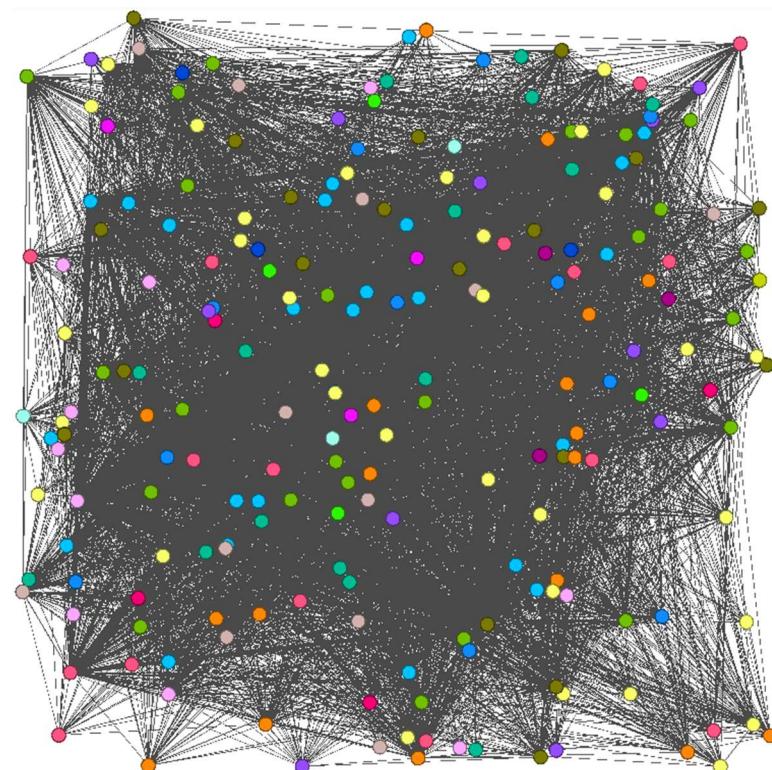
Se le conoce como subtipo o segundo tipo a un tipo secundario que puede desarrollar una especie de Pokémon. El subtipo normalmente solo afecta al tipo de movimientos que aprende el Pokémon, pero también en una mínima parte a su aspecto físico.

Además, un subtipo puede influir en la efectividad de los movimientos que reciba un Pokémon. Por ejemplo: un Trapinch tiene como debilidad los ataques de tipo hielo, agua y planta, y es resistente a los ataques de tipo roca y veneno. Si Trapinch evoluciona a Vibrava gana un subtipo (que es el tipo dragón), por lo que ahora su debilidad es ante los ataques de tipo dragón, hielo y hada. Se le agregan los tipos dragón y hada por tener de subtipo dragón (que es una debilidad de estos) y se le quitan las debilidades a agua y planta porque su subtipo es resistente a estos dos. Aunque debido a su subtipo, ahora es más débil contra el tipo hielo.

Sabiendo esto es curioso que el subtipo más común es veneno ya que esto hace que muchos de los pokemons de esta temporada sean capaces de aprender los movimientos de tipo veneno los cuales son muy útiles. El subtipo también cambia la efectividad y debilidad de estos, por lo que también es interesante saber que al tipo veneno sólo le hacen “counter” el tipo psíquico y tipo tierra, él es efectivo contra tipo bicho, hada, lucha, planta, veneno, por lo que no parece para nada mala opción como subtipo.

tipo2	
Volador	(14,95 %)
Agua	(11,21 %)
Veneno	(11,21 %)
Normal	(8,88 %)
Tierra	(8,41 %)
Fuego	(7,48 %)
Psíquico	(7,01 %)
Lucha	(5,61 %)
Eléctrico	(5,61 %)
Planta	(5,14 %)
Hada	(4,67 %)
Acero	(1,87 %)
Roca	(1,87 %)
Siniestro	(1,4 %)
Bicho	(1,4 %)
Hielo	(1,4 %)
Dragón	(1,4 %)
Fantasma	(0,47 %)

#### Gephi: Partición por atributo



En esta quinta temporada nos damos cuenta de que el subtipo predominante es Volador, esto es bastante útil ya que estos pókemons van a poder aprender ataques de este tipo, como es “vuelo” que permite que los usuarios monten a la espalda de sus pokémon.

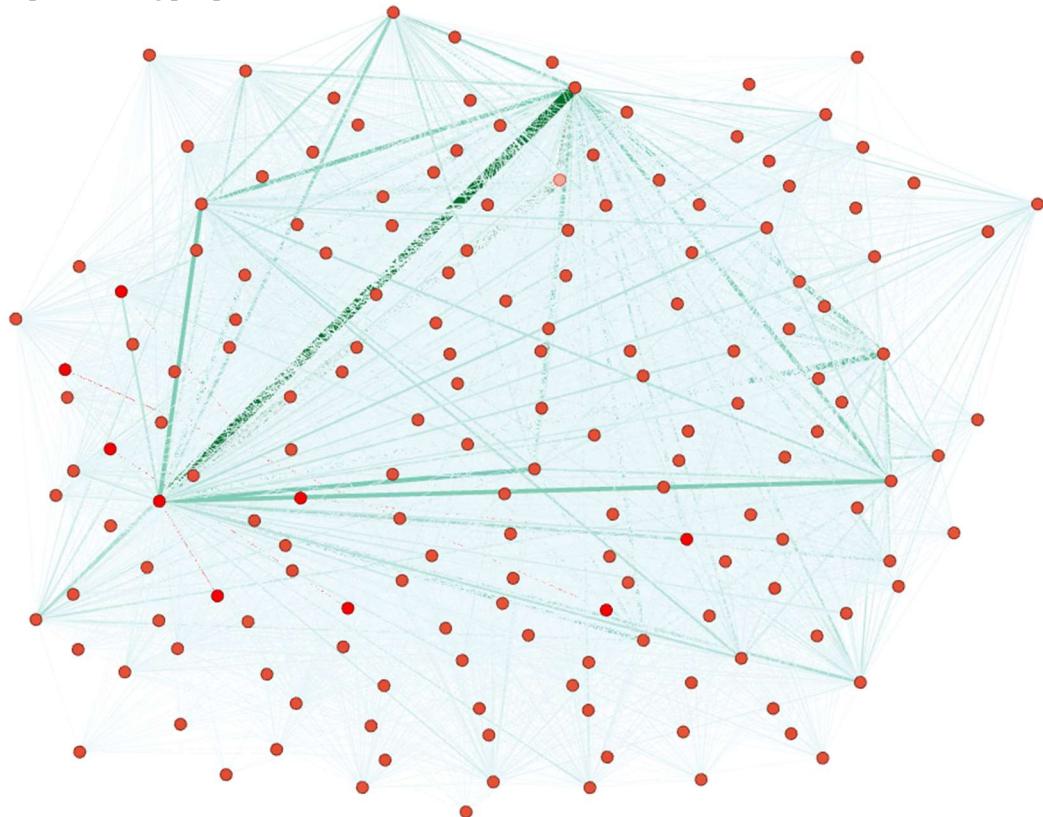
Respecto a la eficacia y debilidad de este tipo tenemos que es débil contra tipos eléctrico, hielo y roca, siendo al contrario eficaz contra los tipos bicho, lucha y planta.

Es interesante darnos cuenta que en ambas temporadas el segundo subtipo más numeroso es agua, haciendo este tipo el predominante en los pokemons.

En la siguiente imagen hacemos un ranking por peso de las aristas, para hacer notar las que mayor peso que ya os adelanto que son la de los pokemons que aparecen en todos los episodios de las temporadas:

T2:

Gephi: Ranking por peso

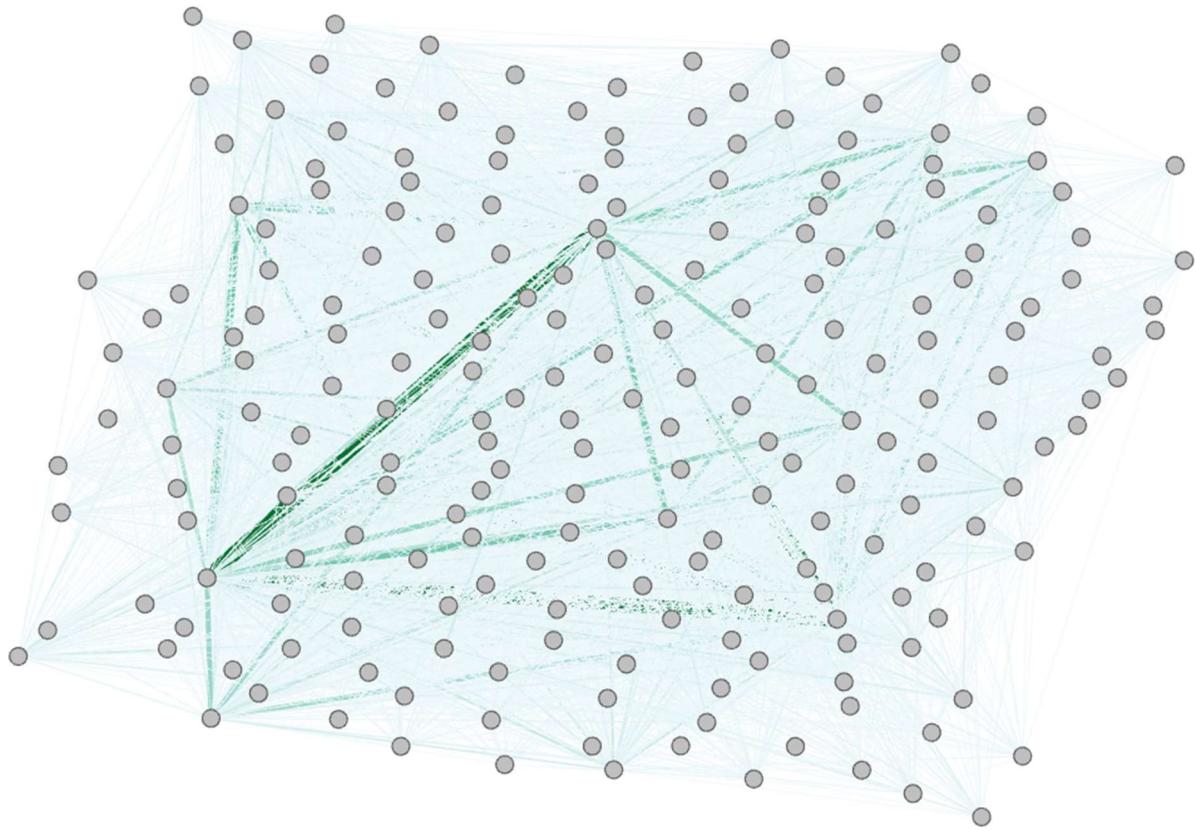


Como podemos ver las aristas que más resaltan por diferencia son las de “Togepi”, “Pikachu”, “Meowth” aunque hay tantas aristas que las de “Pikachu” no se aprecian demasiado bien por estar superpuestas por aristas más claras. Seguidas por el próximo pokémon que ha aparecido en 35 episodios, “Squirtle”.

Origen	Destino	Tipo	Id	Label	Interval
29 - Pikachu	67 - Meowth	No dirigida	5424	63.0	
29 - Pikachu	165 - Togepi	No dirigida	5522	63.0	
67 - Meowth	165 - Togepi	No dirigida	6216	63.0	
29 - Pikachu	6 - Squirtle	No dirigida	5364	35.0	

T5:

En esta temporada los pokemons que destacan son “Pikachu” y “Meowth”. Luego tienen detrás por tan solo un capítulo a “Wobbuffet” y a “Togepi” como podemos ver en la siguientes imágenes;



Gephi: Ranking por peso

Origen	Destino	Tipo	Id	Label	Interval
18 - Pikachu	38 - Meowth	No dirigida	10478		65.0
18 - Pikachu	138 - Togepi	No dirigida	10578		64.0
18 - Pikachu	160 - Wobb...	No dirigida	10600		64.0
38 - Meowth	138 - Togepi	No dirigida	10911		64.0
38 - Meowth	160 - Wobb...	No dirigida	10933		64.0

Como podemos apreciar al tener pokemons que salen en todos los episodios todos los pokémons van a tener como mínimo una arista con valor igual al número de episodios en los que ese pokémon aparece.

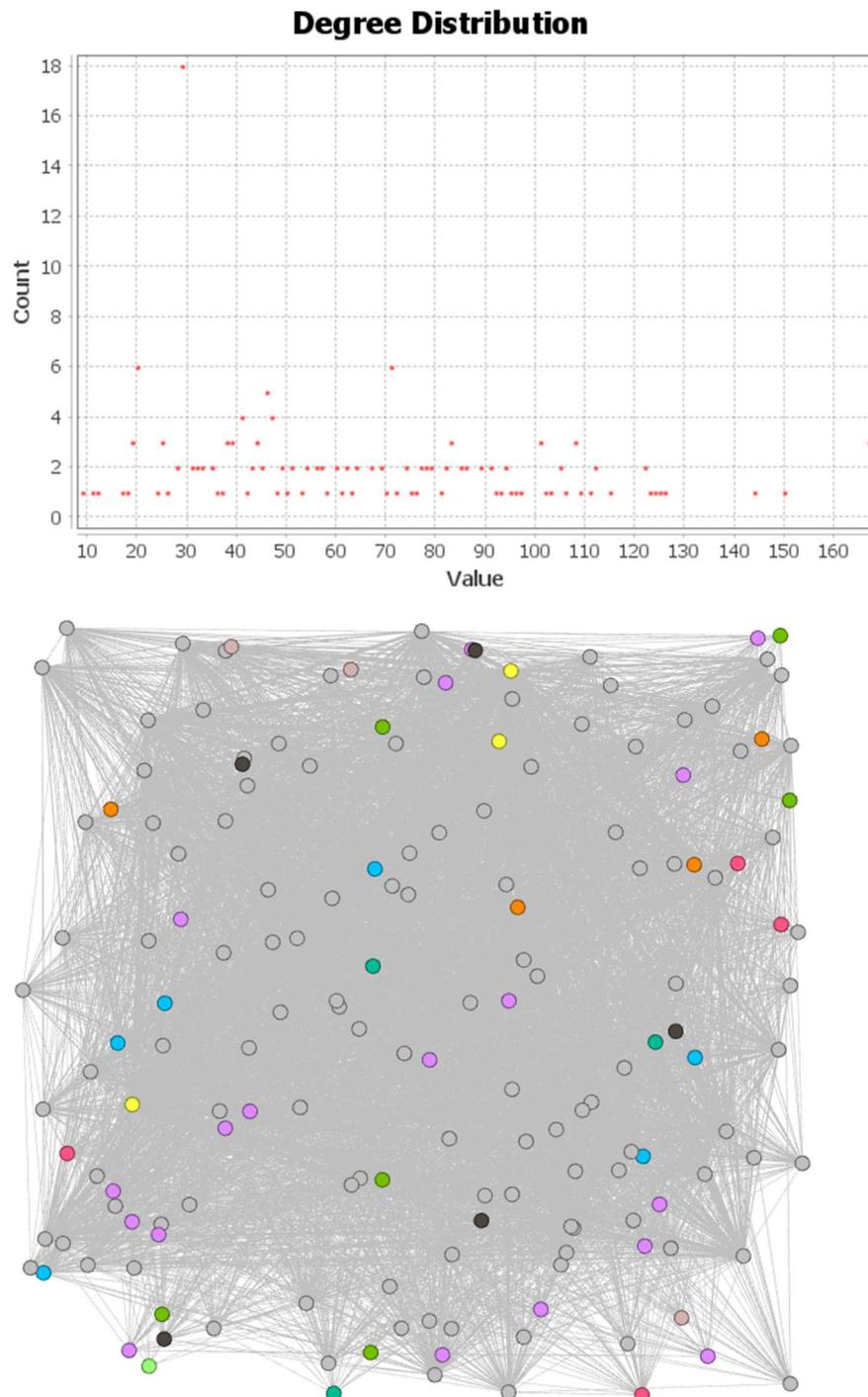
Ahora vamos a analizar el grado medio del grafo y a comentar los nodos con mayor grado y los de menor para ver cuales son los más importantes y cuáles menos, antes de hacer nada podemos saber por adelantado que los nodos con mayor grado van a ser los pokemons que aparecen en todos los episodios de las temporadas, ya que estos se relacionan con todos los pokemons que aparecen en esta. Esto nos da una idea de qué nodos son los más “sociables” y cuáles no son demasiado importantes.

## Degree Report | T2

---

## Results:

Average Degree: 61,857



Gephi: Partición por grado

El grado medio de los nodos de este grafo es aproximadamente 62, esto quiere decir que los pokemons de este estudio de media han aparecido junto a 62 pokemons a lo largo del estudio.

Aquí vemos los coloreados de amarillo los pókemons que más grado tienen, constan de grado 167 y son los ya mencionados antes que aparecen en toda la temporada, están

seguidos por “Arbok” con 150 apareciendo en tan solo 26 episodios. Por otra parte coloreado de verde clarito abajo a la izquierda tenemos a “Magneton” que consta de grado 9, lo que es bastante poco, le sigue con grado 11 “Kabuto”.

El grado medio de esta temporada es 61,857 esto es bastante elevado teniendo en cuenta que esta red consta de 168 nodos esto representa un 36,8% de los pokemons, esto nos da a entender la gran densidad de este grafo en el que hay una gran conectividad. Luego podemos darnos cuenta de que el grado más numeroso de lejos es 29 que lo comparten un 10,71% que se traduce en 18 nodos, estos están pintados de morado.

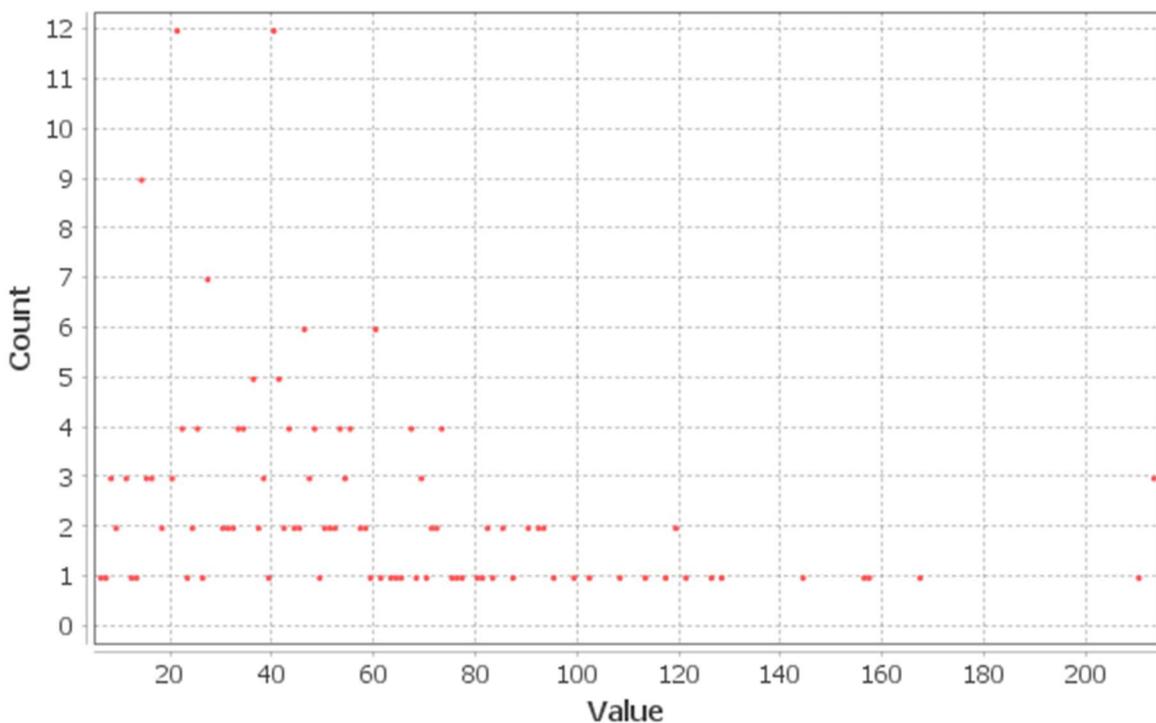
## Degree Report | T5

---

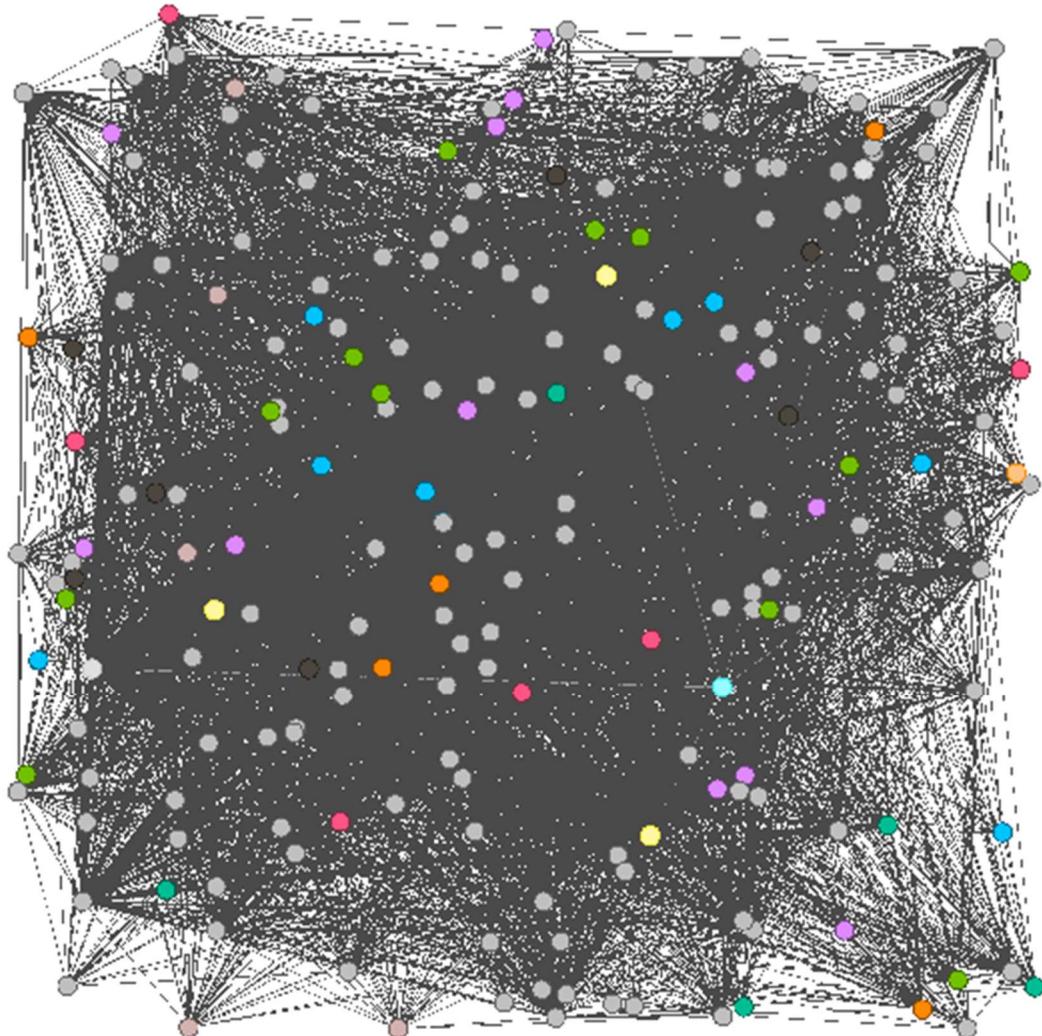
### Results:

Average Degree: 51,159

**Degree Distribution**



En esta temporada el grado medio de esta red es 51,159 lo que representa un 23,9% en comparación a la anterior temporada este número es bastante bajo, teniendo en cuenta que en la temporada anterior el grado medio era mayor y eso que aparecían menos pokemons en la temporada. Comparando esta temporada que consta de 65 episodios y 5474 aristas con la temporada 2 que tiene 63 y 5196 aristas sabiendo que en la temporada 2 tiene un 78,5% de los nodos de la temporada 5 y aún así consta de 94,92% de sus aristas, esto nos dice que la temporada dos es mucho más densa y que aparecen casi los mismos pokemons por episodio.



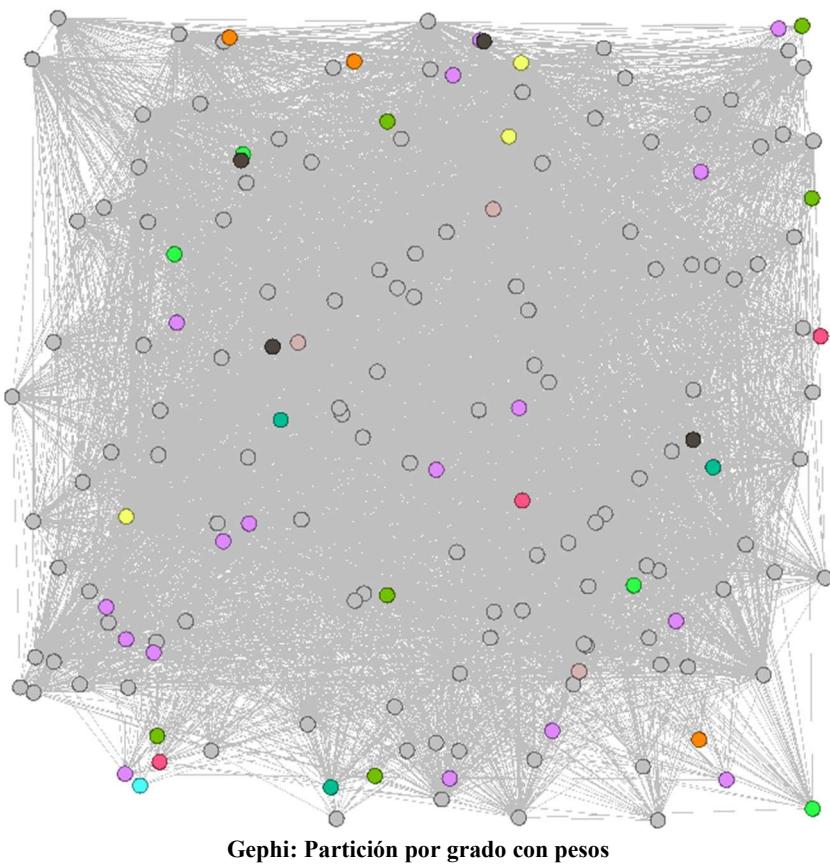
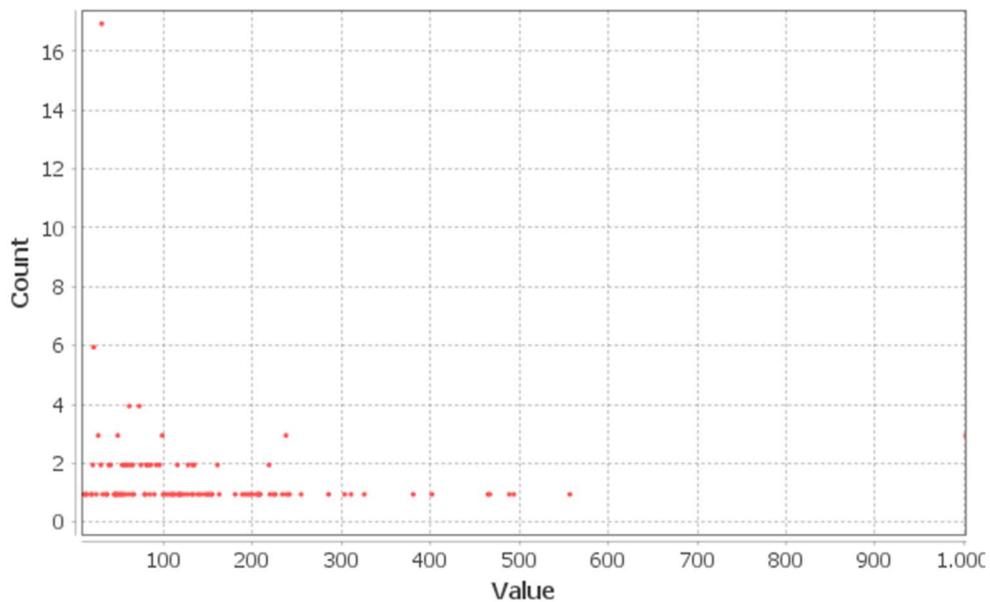
Aquí podemos observar a los nodos con mayor grado de la red pintados de amarillo, aquí están empatados los tres pokemons que aparecen todos los episodios de la temporada, estos son; “Pikachu”, “Meowth” y “Wobbuffet” que constan de grado 213, estos ganadores son seguidos por Togepi con 210. Por el otro lado mirando los nodos con menor grado tenemos un claro ganador que está pintado de azul clarito y representa a “Xatu” que tan sólo aparece en el episodio 236 donde aparecen tan solo 6 pokemons aparte de él. Luego es interesante contemplar que los grados más numerosos están pintados de morado y de verde que tienen 21 y 40 respectivamente, cada uno de estos grupos cuentan con 5,61% de los nodos (12).

## Weighted Degree Report | T2

### Results:

Average Weighted Degree: 126,714

### Degree Distribution



Gephi: Partición por grado con pesos

Aquí vemos el grafo coloreado según su grado teniendo en cuenta los pesos de las aristas de los nodos, aquí siguen en cabeza para variar los mismos nodos de siempre; “Togepi”, “Pikachu” y “Meowth” esta vez con un valor de 1000, están pintados de amarillo, detrás de estos les sigue “Squirtle” con 555 y grado 125 es curioso compararlo con “Arbok” que es el que mayor grado tiene después de los 3 anteriormente mencionados con 150 pero un grado con peso de 463 esto nos quiere decir que aunque “Squirtle” haya aparecido en episodios junto a menos pokemons, ha coincidido con estos mismos pokemons en varios

episodios haciendo que aumente el peso de sus aristas y consiguiendo superar a “Arbok” por consecuencia. El nodo con menor grado con peso es el azul clarito que representa a “Magneton” y que tiene el mismo grado con peso que sin peso, 9, esto se debe a que tan sólo aparece en un episodio por lo que el valor de todas sus aristas es 1. El grupo de nodos más numeroso que comparten grado con peso están pintados de morado, tienen grado con peso de 29 y son el 10,12% del total de los nodos (17 nodos). Este fenómeno anteriormente mencionado me ha llamado la atención por lo que al ir a mirar al laboratorio de datos me he dado cuenta de que todos los pokemons de este grupo también tienen el mismo grado, 29, esto nos sugiere al tener el mismo grado con peso y el mismo grado que salen en tan sólo un capítulo.

Id	Label	...	pokedex	pokemon	tipo1	tipo2	Grado
87	BellsproutIR	69	BellsproutIR	Planta	Veneno	29	29.0
74	PrimeapeIR	57	PrimeapeIR	Lucha	Lucha	29	29.0
72	MankeyIR	56	MankeyIR	Lucha	Lucha	29	29.0
65	DiglettIR	50	DiglettIR	Tierra	Tierra	29	29.0
62	VenonatIR	48	VenonatIR	Bicho	Veneno	29	29.0
59	ParasIR	46	ParasIR	Bicho	Planta	29	29.0
56	VileplumeIR	45	VileplumeIR	Planta	Veneno	29	29.0
52	OddishIR	43	OddishIR	Planta	Veneno	29	29.0
43	NidokingIR	34	NidokingIR	Veneno	Tierra	29	29.0
40	Nidoran♂IR	32	Nidoran♂IR	Veneno	Veneno	29	29.0
35	Nidoran♀IR	29	Nidoran♀IR	Veneno	Veneno	29	29.0
19	PidgeyIR	16	PidgeyIR	Normal	Volador	29	29.0
15	WeedleIR	13	WeedleIR	Bicho	Veneno	29	29.0
133	RhydonIR	112	RhydonIR	Tierra	Roca	29	29.0
131	RhyhornIR	111	RhyhornIR	Tierra	Roca	29	29.0
122	ExegutorIR	103	ExegutorIR	Planta	Psíquico	29	29.0
10	CaterpieIR	10	CaterpieIR	Bicho	Bicho	29	29.0

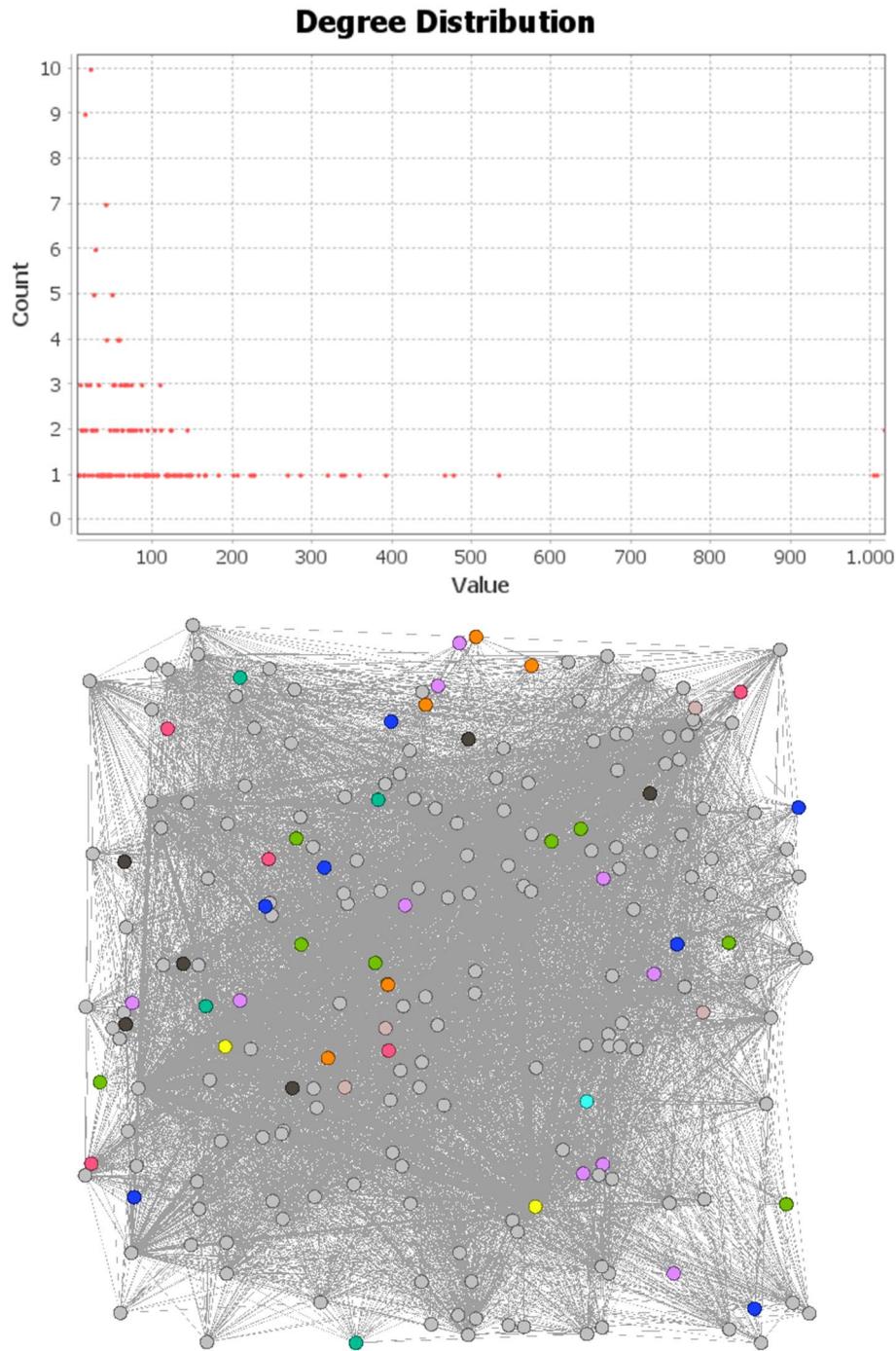
Aquí rápidamente revisando la base de datos me di cuenta de que todos estos pokemons pertenecían a un capítulo en particular el EP090 en este episodio los protagonistas de nuestra aventura van a la Isla Rosada por ello detrás de todos los nombres de los pokemons que vemos en la imagen anterior tienen IR las iniciales de esta isla, estos pokemons están diferenciados a sus equivalentes continentales ya que los pokemons de esta isla tienen un color diferente (son de color rosado) y por ello en “Wikidex” nuestra página de referencia los ponen separados. Este episodio posiblemente sea el episodio donde más pokemons salen de la temporada contando con 30 de estos y 16 exclusivos de este episodio,

## Weighted Degree Report | T5

---

### Results:

Average Weighted Degree: 95,514



#### Gephi: Partición por grado con pesos

Aquí vemos el grafo coloreado según su grado teniendo en cuenta los pesos de las aristas de los nodos, aquí siguen en cabeza para variar los mismos nodos de siempre; “Pikachu” y “Meowth” que constan de un grado con pesos de 1017 les sigue de cerca “Wobbuffet” con 1007 luego en el lado contrario el nodo con menor grado con peso es nuevamente “Xatu” con un grado con peso equivalente a su grado normal debido a que sólo aparece en un episodio, este grado es 6, está pintado en el grafo de azul claro.

El grupo más numeroso que comparten grado con peso contiene un 4,67% del total de los nodos de esta temporada, 10, estos están pintados de morado y tienen un grado con peso

de 21. Al fijarme más a fondo en estos nodos me percaté de que todos compartían grado y grado con peso como podéis observar en la siguiente imagen.

Id	Label	...	pokedex	pokemon	tipo1	tipo2	Grado	...
197	Suicune		245	Suicune	Agua	Agua	21	0 21.0
195	Raikou		243	Raikou	Eléctrico	Eléctrico	21	0 21.0
187	Donphan		232	Donphan	Tierra	Tierra	21	0 21.0
164	Gligar		207	Gligar	Tierra	Volador	21	0 21.0
132	Ariados		168	Ariados	Bicho	Veneno	21	0 21.0
80	Electrode		101	Electrode	Eléctrico	Eléctrico	21	0 21.0
75	Gengar		94	Gengar	Fantas...	Veneno	21	0 21.0
73	Gastly		92	Gastly	Fantas...	Veneno	21	0 21.0
49	Alakazam		65	Alakazam	Psíquico	Psíquico	21	0 21.0
30	Wigglytuff		40	Wigglytuff	Normal	Hada	21	0 21.0

Esto me sugiere que todos han aparecido en un solo episodio todos juntos este episodio es el EP229 en el que aparecen 22 pokemons y 10 exclusivos de este episodio.

## Graph Distance Report | T2

---

### Parameters:

Network Interpretation: undirected

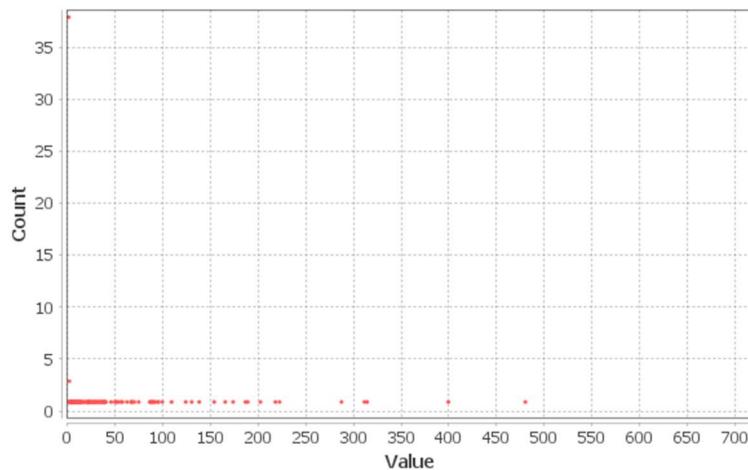
### Results:

Diameter: 2

Radius: 1

Average Path length: 1.629669232962646

#### **Betweenness Centrality Distribution**

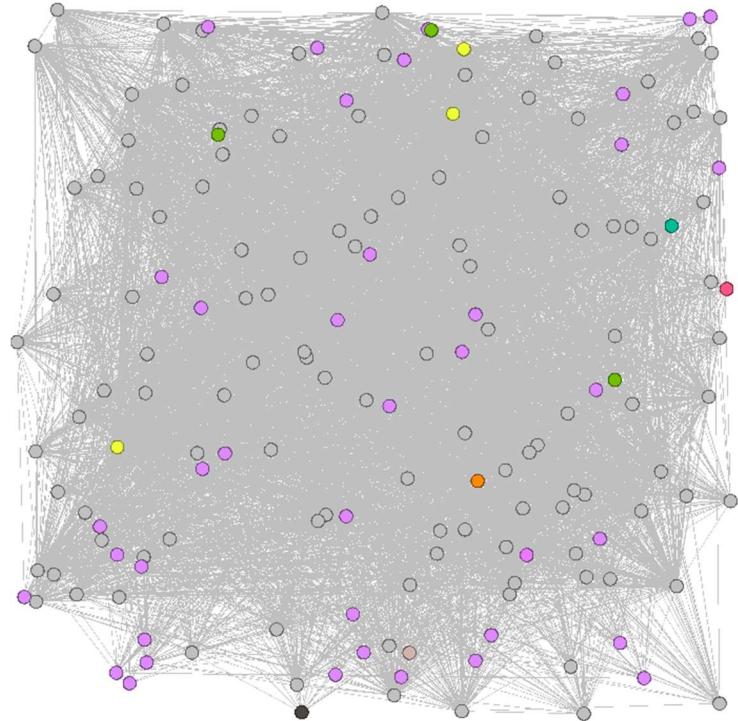


En la teoría de grafos, la centralidad de intermediación es una medida de centralidad en un gráfico basada en los caminos más cortos. Para cada par de vértices en un grafo conectado, existe al menos un camino más corto entre los vértices tal que el número de aristas por las que pasa el camino (para gráficos no ponderados) o la suma de los pesos de las aristas (para grafos ponderados) se minimiza.

La centralidad de intermediación para cada vértice es el número de estos caminos más cortos que pasan por el vértice.

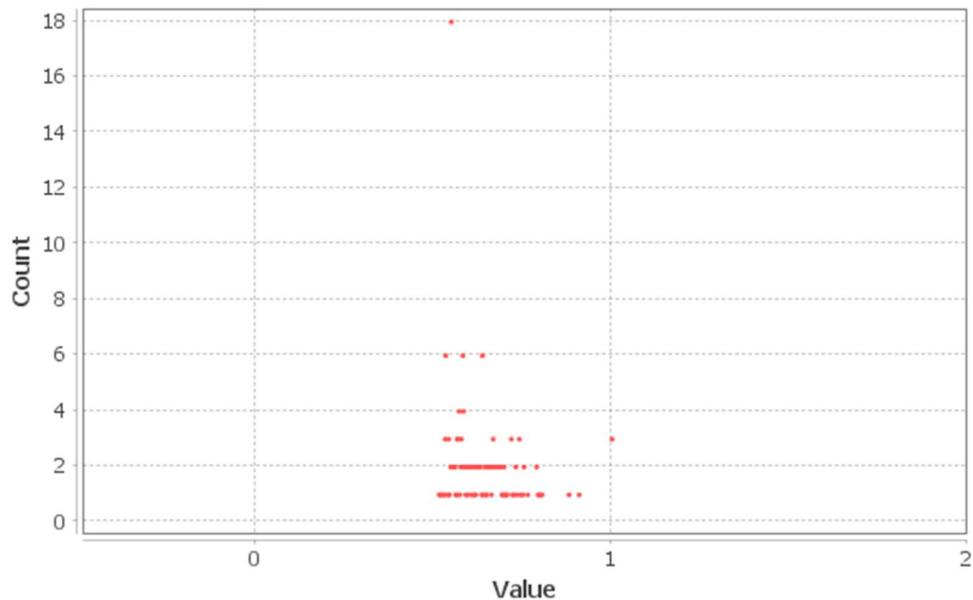
En el grafo mostrado más abajo podemos darnos cuenta de que nodos obtienen la mayor centralidad de intermediación estos son los de color gris, y entre ellos destaca los nodos “Togepi”, “Pikachu” y “Meowth” que en la figura se reconoce por ser amarillos que cuentan con 717,58 de centralidad de intermediación. Es lógico que estos nodos sean los que tienen centralidad más alta ya que al conectarse con el resto de nodos todos los caminos mínimos de más de 1 van a pasar por ellos, ya que si desde un nodo “a” queremos llegar a un nodo “b” y estos no están conectados, la forma más rápida de llegar a ese nodo “b” es pasar por uno de estos 3 nodos logrando así un camino de 2 como máximo para cualquier camino del grafo.

Por otro lado de morado tenemos pintados los nodos con 0.0 que representan el 22,62% de los nodos, 38 elementos, este grupo es tan numeroso debido a que hay muchos nodos por los que no va a pasar ningún camino mínimo ya que no es necesario..



Gephi: Partición por betweenness centrality

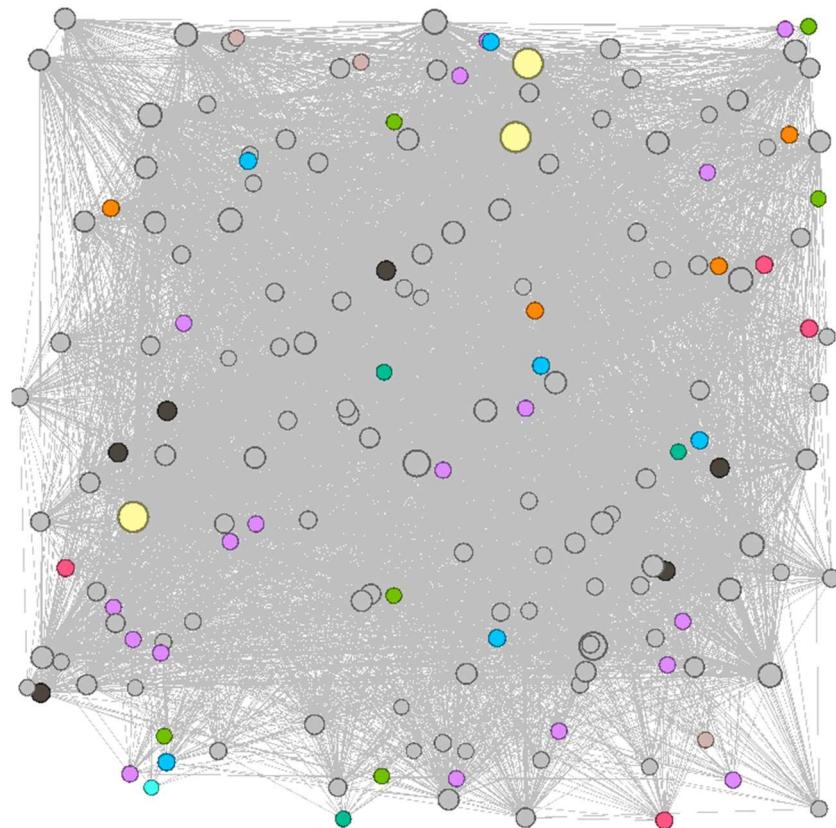
### Closeness Centrality Distribution



En un gráfico conectado, la centralidad de cercanía (o cercanía) de un nodo es una medida de centralidad en una red, calculada como la suma de la longitud de los caminos más cortos entre el nodo y todos los demás nodos en el gráfico. Por lo tanto, cuanto más central es un nodo, más cerca está de todos los demás nodos.

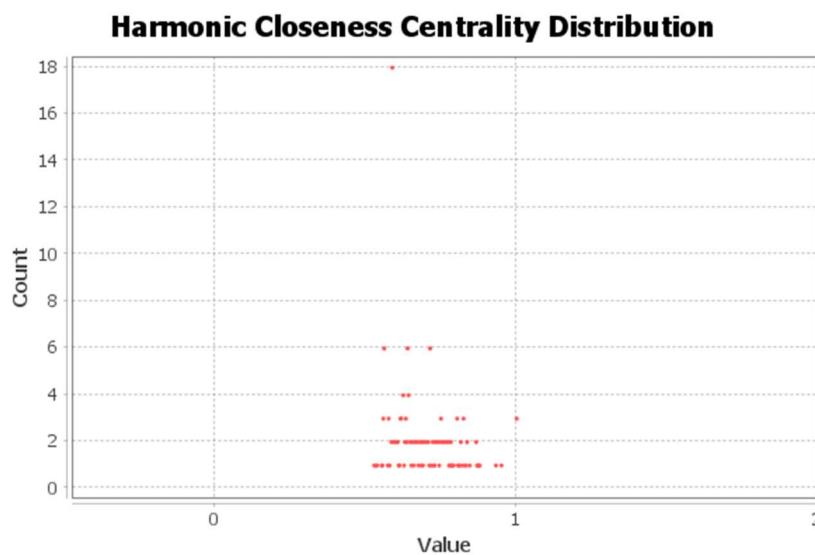
En la figura que encontramos más abajo podemos reconocer los nodos con mayor centralidad de cercanía en un ranking de mayor a menor tamaño; entre los de mayor tamaño destacan los amarillos, que son el grupo con la mayor centralidad de cercanía y a la vez es la centralidad que más se repite entre los nodos.

Estos nodos tienen caminos realmente cortos entre el resto de los nodos con los que puede conectarse. Ósea que se encuentren en su misma componente conexa ya que en esta red solo tiene una componente conexa. Los nodos amarillos tienen 1 de centralidad de cercanía debido a que su camino mínimo para todos los nodos es 1 debido a que están enlazados con todos los demás nodos. Luego los nodos pintados de morados es el número más numeroso, este es así debido al episodio de la Isla Rosada nuevamente ya que todos estos pokemons comparten completamente los valores de las métricas, excepto “Slowbro” que ha dado la casualidad que tiene la misma centralidad de cercanía debido a sus características propias (todo esto podéis contrastarlo en la segunda imagen de abajo). Por último es curioso mirar a “Magneton” de nuevo abajo a la izquierda pintado de azul clarito ya que es el que menor centralidad de cercanía con 0.5138 este coincide con ser el nodo con menor grado.



Gephi: Partición por closeness centrality

WeedleIR	0.547541
CaterpieIR	0.547541
PidgeyIR	0.547541
Nidoran♀IR	0.547541
Nidoran♂IR	0.547541
NidokingIR	0.547541
OddishIR	0.547541
VileplumeIR	0.547541
ParasIR	0.547541
VenonatIR	0.547541
DiglettIR	0.547541
MankeyIR	0.547541
PrimeapeIR	0.547541
BellsproutIR	0.547541
Slowbro	0.547541
ExeggutorIR	0.547541
RhyhornIR	0.547541
RhydonIR	0.547541

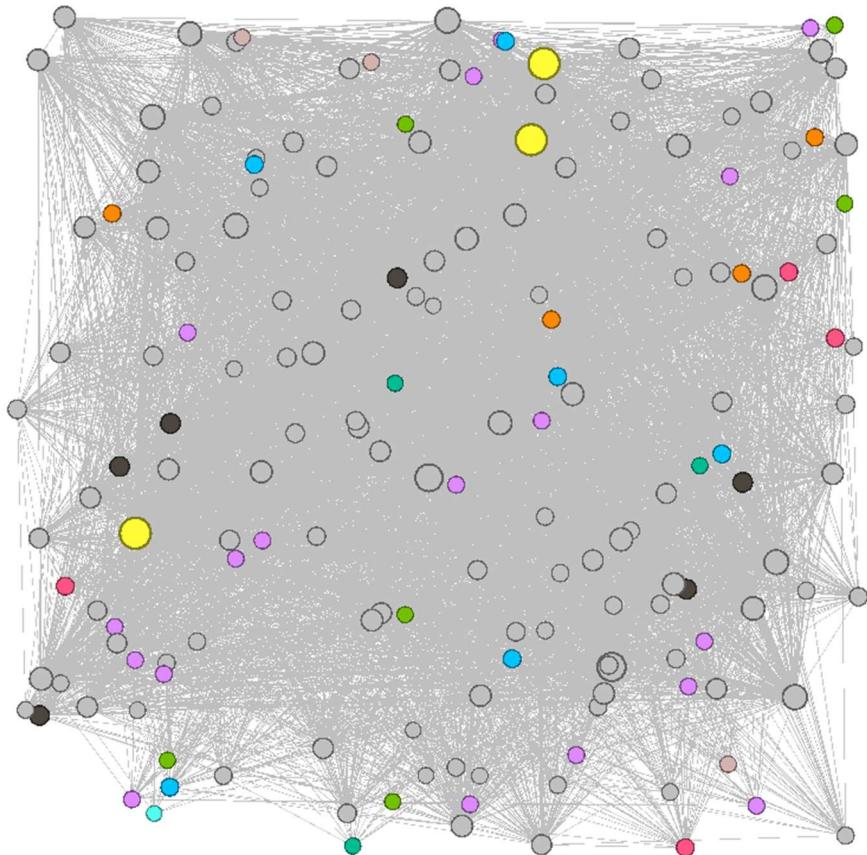


La centralidad armónica (también conocida como centralidad valorada) es una variante de la centralidad de cercanía, que se inventó para resolver el problema que tenía la fórmula original al tratar con grafos no conectados. Como ocurre con muchos de los algoritmos de centralidad, se origina en el campo del análisis de redes sociales.

Podemos darnos cuenta de que los nodos con centralidad armónica 1 (los amarillos) son los que estaban en amarillo en la anterior figura (tenían centralidad de cercanía 1).

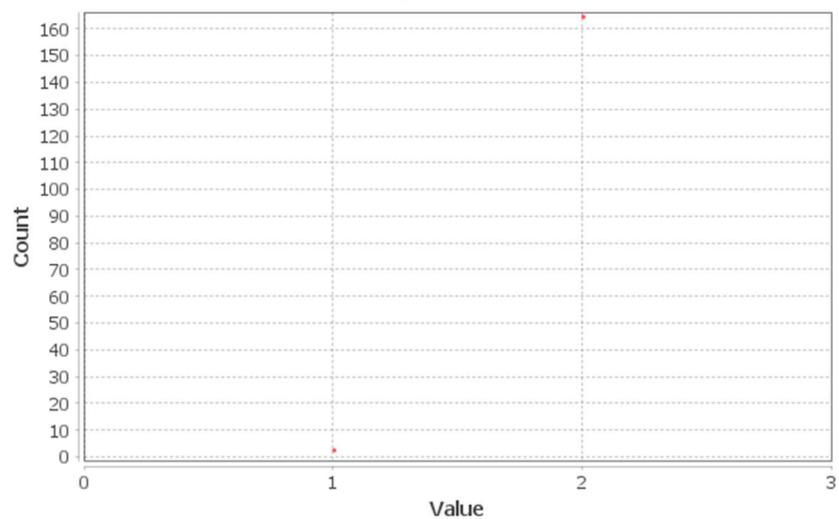
El de menor centralidad armónica es el pintado de azul que vuelve a ser “Magneton” que es el que tiene menor grado.

El grupo más numeroso son los nodos pintados de morado que cuentan con 18 pokemons en ese grupo.



Gephi: Partición por harmonic closeness centrality

### Eccentricity Distribution



## Algorithm:

Ulrik Brandes, *A Faster Algorithm for Betweenness Centrality*, in Journal of Mathematical Sociology 25(2):163-177, (2001).

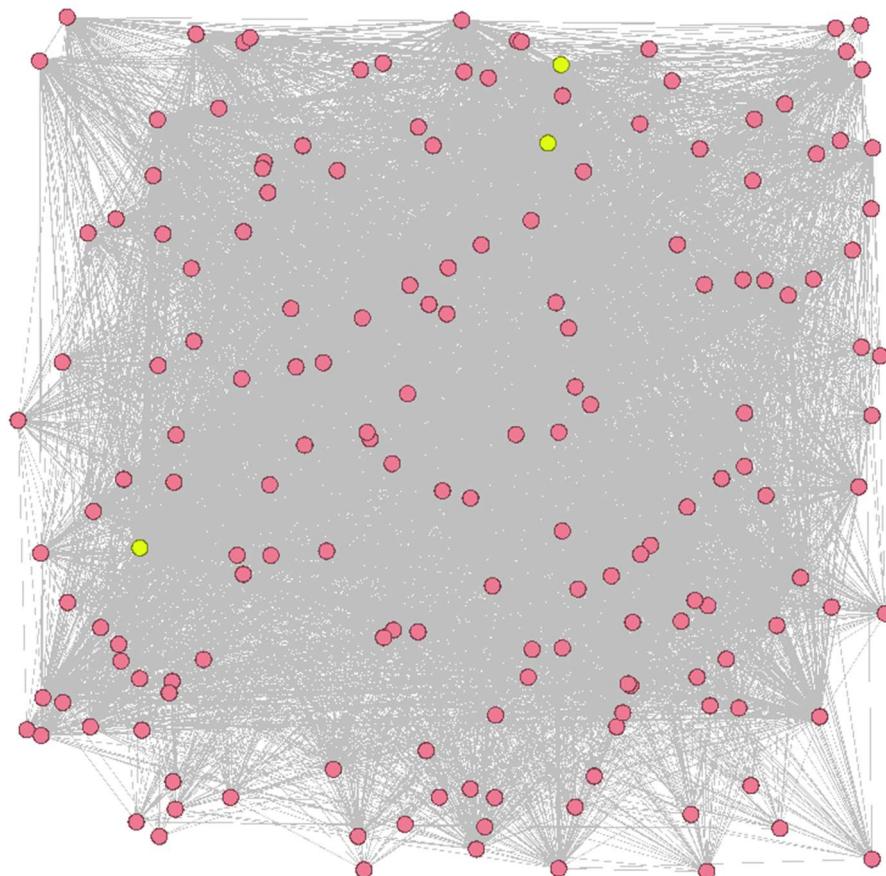
Como podemos observar la mayor distancia entre un nodo y otro es 2, la gran mayoría de nodos tienen una excentricidad de 2, excepto los 3 pokemons que tienen grado máximo tienen una excentricidad de 1.

Gephi en este caso nos ofrece datos interesantes a valorar como es el diámetro de la red, que corresponde con el camino más largo del grafo, en este caso es de 2, siendo un grafo con 168 nodos es poquísimo, esta anomalía se debe a los 3 nodos que conectan con todos los demás, que sirven como puente.

Si el diámetro es la máxima excentricidad del grafo, el radio es la menor. En este caso vemos que es de 1 ya que los tres pokemons protagonistas están a 1 camino de todo el resto de nodos.

También es curioso fijarnos que el camino medio es 1,62.

En la siguiente figura podemos fijarnos por tamaño quiénes son los más excéntricos a mayor tamaño, mayor excentricidad. Los nodos están agrupados por colores según su excentricidad, siendo amarillos los que menos nodos tienen esa excentricidad, los rojos que son los que tienen excentricidad 2 que es la máxima.



Gephi: Partición por eccentricity

# Graph Distance Report | T5

---

## Parameters:

Network Interpretation: undirected

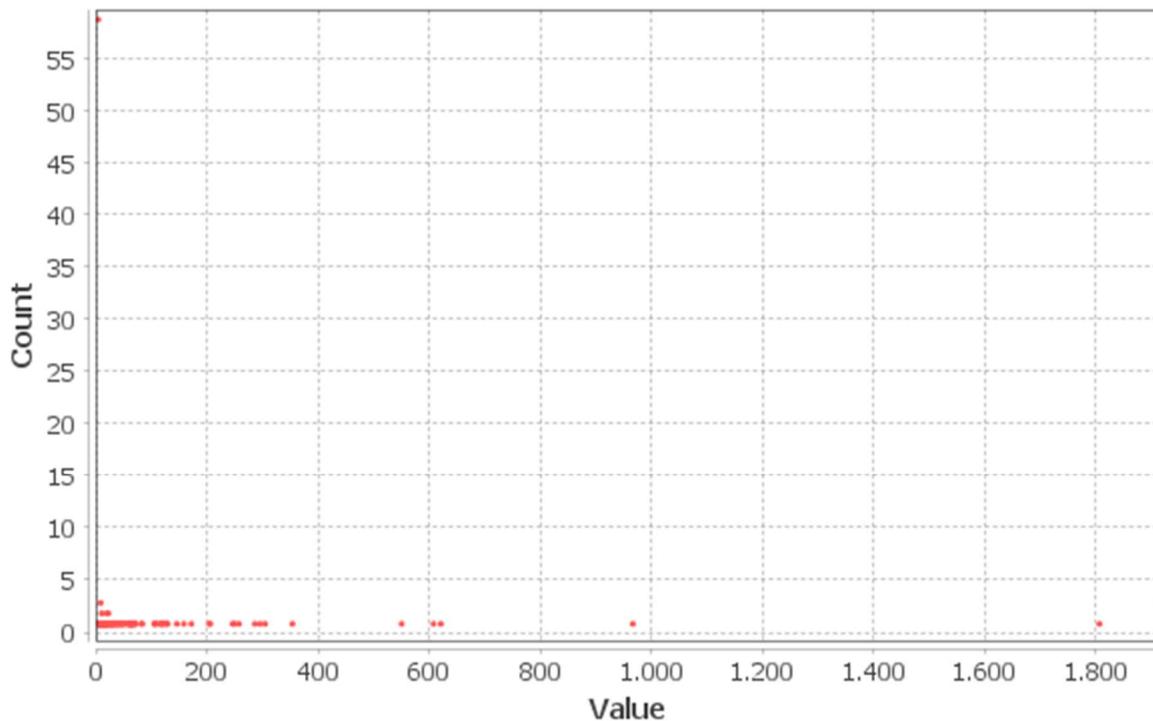
## Results:

Diameter: 2

Radius: 1

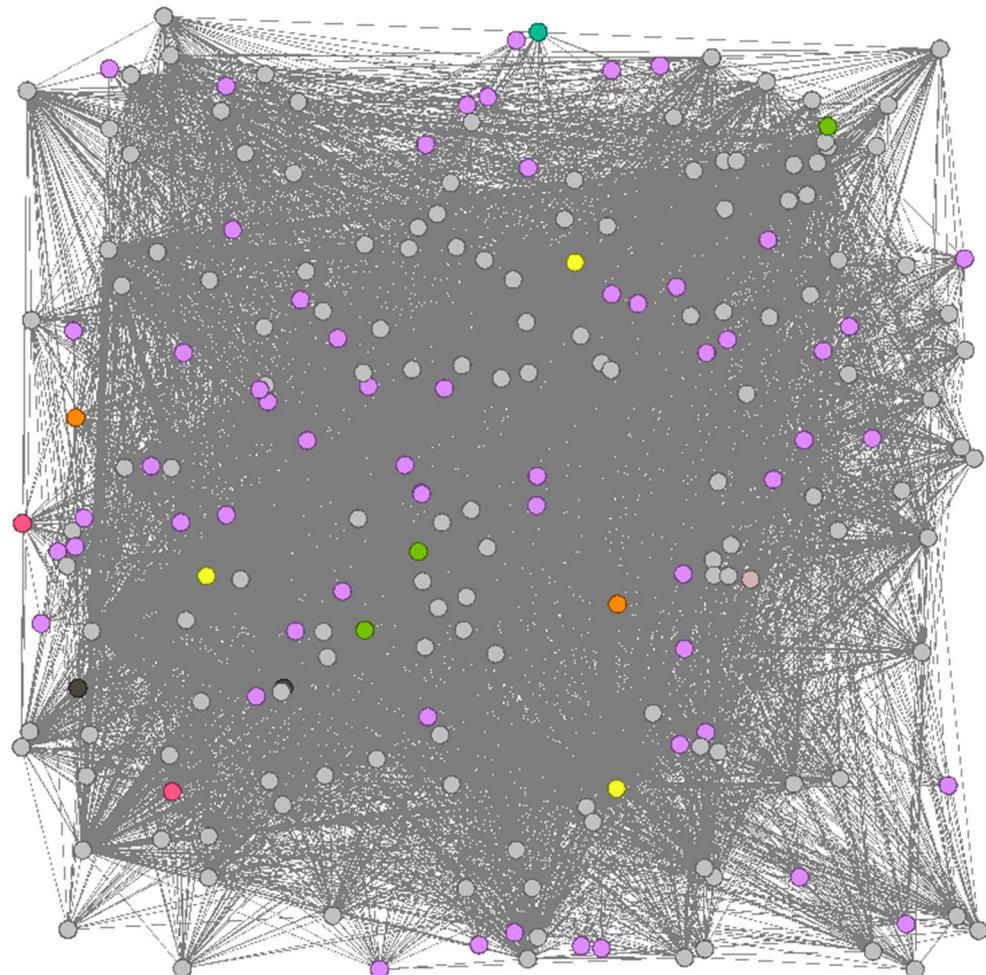
Average Path length: 1.7598613487780264

**Betweenness Centrality Distribution**

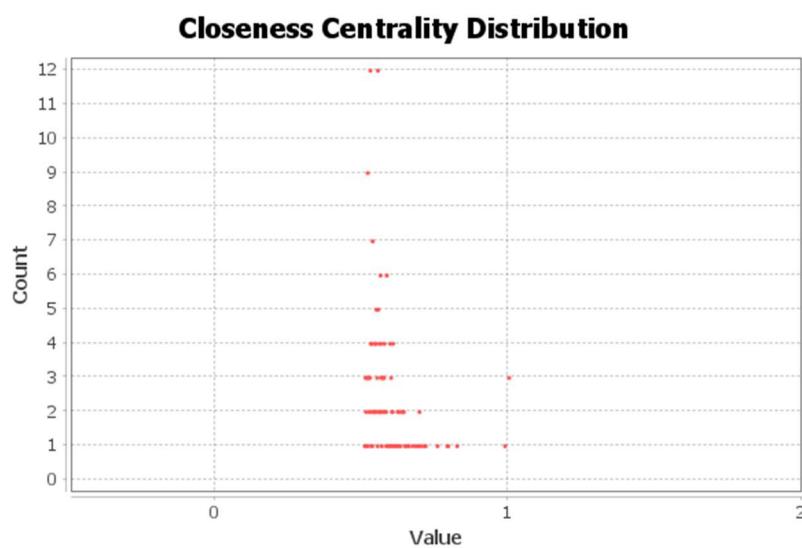


En el grafo mostrado más abajo podemos darnos cuenta de que nodos obtienen la mayor centralidad de intermediación estos son los de color gris, y entre ellos destaca los nodos “Wobbuffet”, “Pikachu” y “Meowth”, en la figura se reconoce por ser amarillos y cuentan con 1911.94 de centralidad de intermediación. Es lógico que estos nodos sean los que tienen centralidad más alta ya que al conectarse con el resto de nodos todos los caminos mínimos de más de 1 van a pasar por ellos, ya que si desde un nodo “a” queremos llegar a un nodo “b” y estos no están conectados, la forma más rápida de llegar a ese nodo “b” es pasar por uno de estos 3 nodos logrando así un camino de 2 como máximo para cualquier camino del grafo.

Por el otro lado de morado tenemos pintados los nodos con 0.0 que representan el 27,57% de los nodos, 59 elementos, este grupo es tan numeroso debido a que hay muchos nodos por los que no va a pasar ningún camino mínimo ya que no es necesario.



Gephi: Partición por betweenness centrality

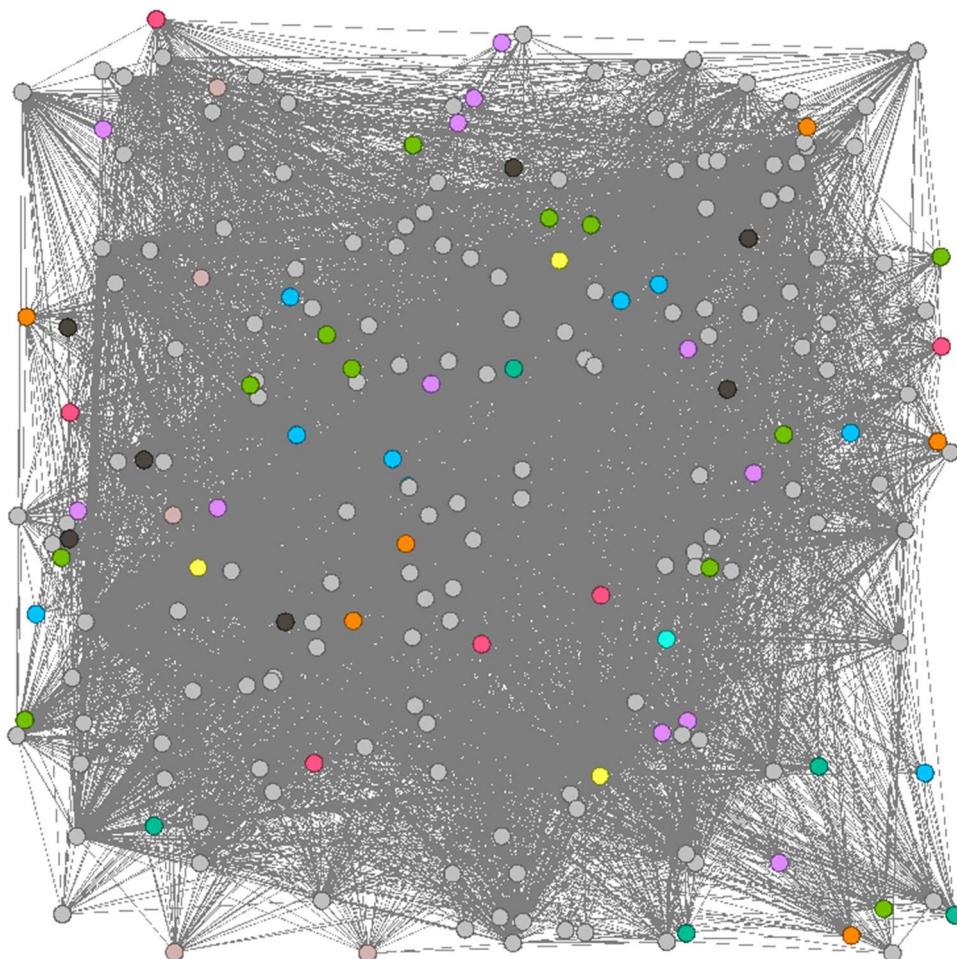


En la figura que encontramos más abajo podemos reconocer los nodos con mayor centralidad de cercanía en un ranking de mayor a menor tamaño; entre los de mayor tamaño destacan los amarillos, que son el grupo con la mayor centralidad de cercanía y a la vez es la centralidad que más se repite entre los nodos.

Estos nodos tienen caminos realmente cortos entre el resto de los nodos con los que puede conectarse. Ósea que se encuentren en su misma componente conexa ya que en esta red solo tiene una componente conexa. Los nodos amarillos tienen 1 de centralidad de cercanía debido a que su camino mínimo para todos los nodos es 1 debido a que están enlazados con todos los demás nodos.

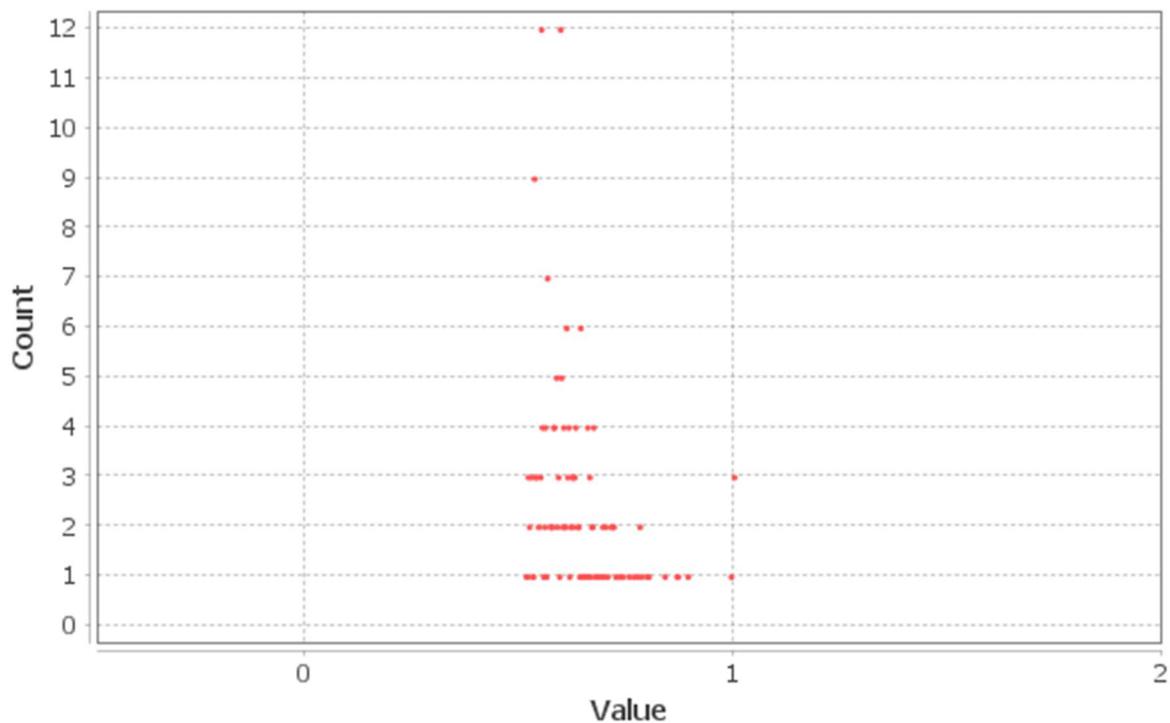
Aquí los nodos amarillos son los mismos que el grafo anterior “Wobbuffet”, “Pikachu” y “Meowth” por las razones anteriormente mencionadas.

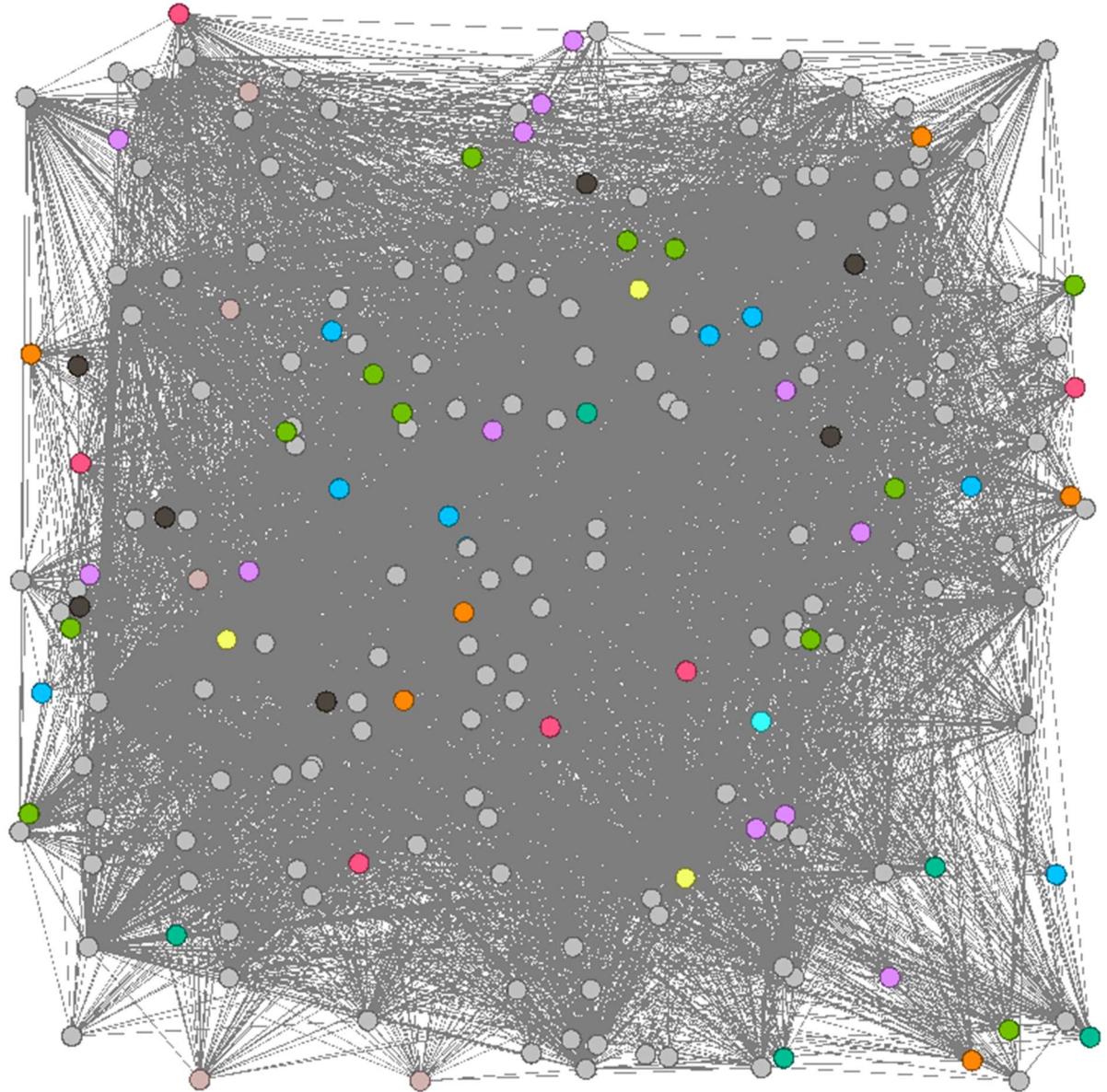
El que tiene menor centralidad de cercanía es igual que antes “Xatu” coloreado de azul claro este coincide con ser el nodo con menor grado de la red. Los grupos más numerosos son morado y verde oscuro con 12 pokemons cada uno.



Gephi: Partición por closeness centrality

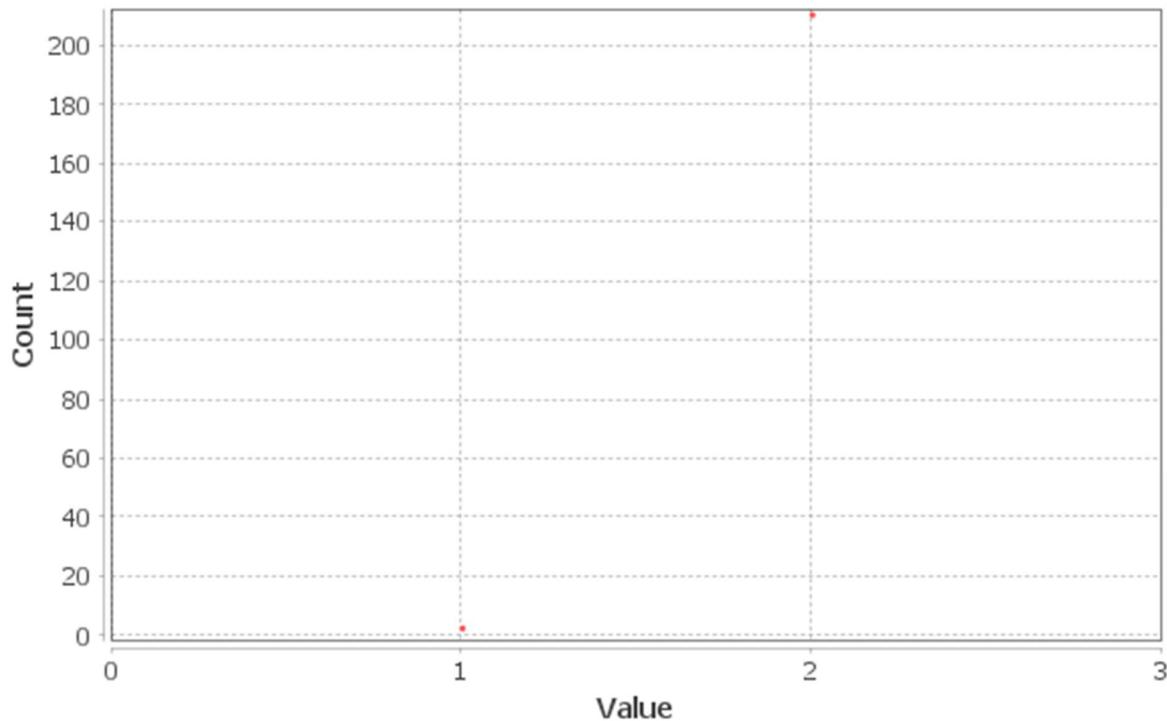
## Harmonic Closeness Centrality Distribution





Gephi: Partición por harmonic closeness centrality

## Eccentricity Distribution



## Algorithm:

Ulrik Brandes, *A Faster Algorithm for Betweenness Centrality*, in Journal of Mathematical Sociology 25(2):163-177, (2001)

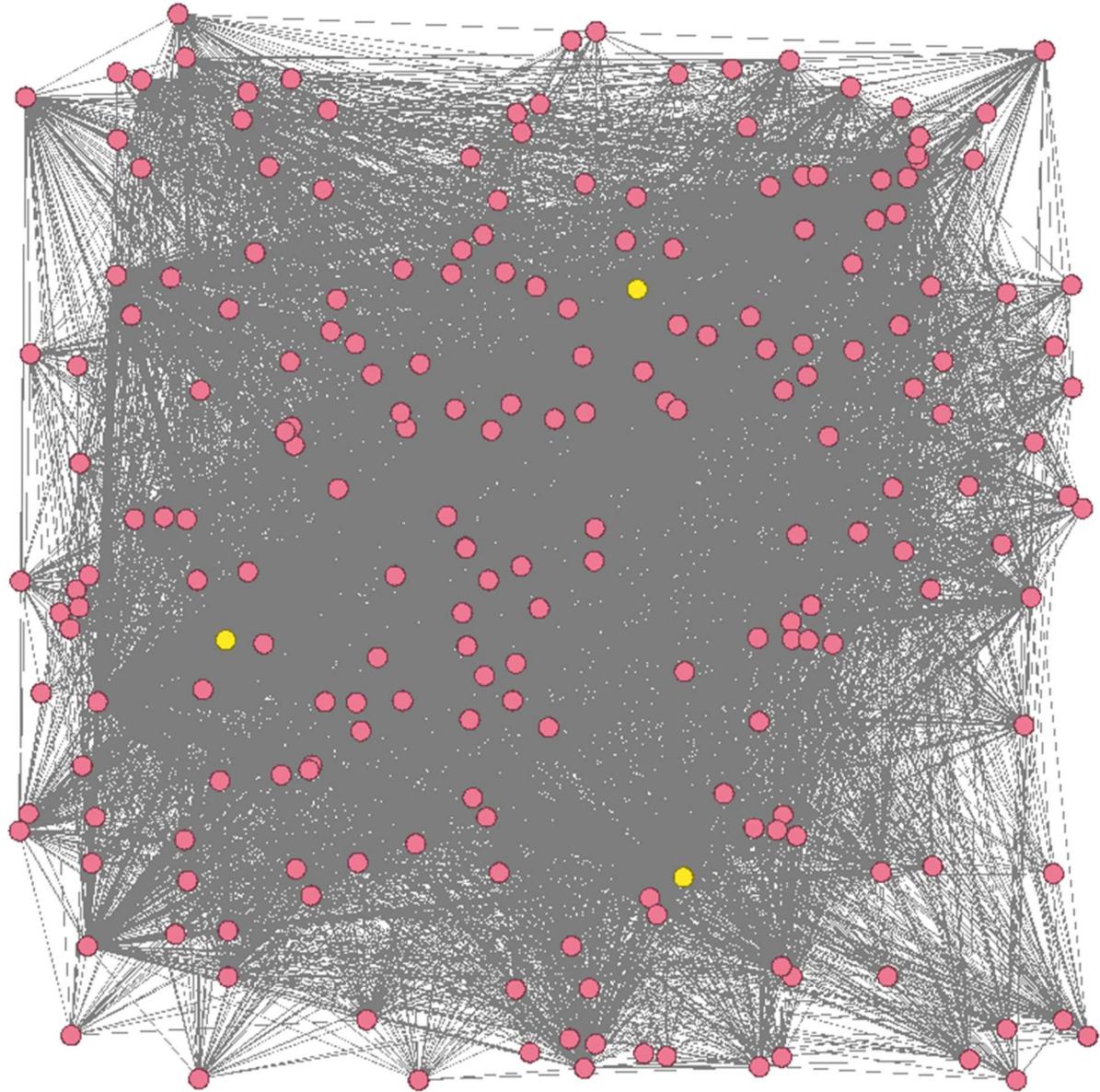
Como podemos observar la mayor distancia entre un nodo y otro es 2, la gran mayoría de nodos tienen una excentricidad de 2, excepto los 3 pokemons que tienen grado máximo tienen una excentricidad de 1.

Gephi en este caso nos ofrece datos interesantes a valorar como es el diámetro de la red, que corresponde con el camino más largo del grafo, en este caso es de 2, siendo un grafo con 168 nodos es poquísimo, esta anomalía se debe a los 3 nodos que conectan con todos los demás, que sirven como puente.

Si el diámetro es la máxima excentricidad del grafo, el radio es la menor. En este caso vemos que es de 1 ya que los tres pokemons protagonistas están a 1 camino de todo el resto de nodos.

También es curioso fijarnos que el camino medio es 1.75.

En la siguiente figura podemos fijarnos por tamaño quiénes son los más excéntricos a mayor tamaño, mayor excentricidad. Los nodos están agrupados por colores según su excentricidad, siendo amarillos los que menos nodos tienen esa excentricidad, los rojos que son los que tienen excentricidad 2 que es la máxima.



Gephi: Partición por eccentricity

# Graph Density Report | T2

---

## Parameters:

Network Interpretation: undirected

## Results:

Density: 0,370 → Esta métrica se calcula fácilmente haciendo comprobando cuantas aristas tendría el grafo si fuese completo → nodos \* grado máximo nodos/2 =  $168 \times 167 / 2 = 14028$  aristas/aristas máximas →  $5196 / 14028 = 0,37$

En teoría de grafos, la densidad de un grafo es una propiedad que determina la proporción de aristas que posee. Un grafo denso es un grafo en el que el número de aristas es cercano al número máximo de aristas posibles, es decir, a las que tendría si el grafo fuera completo. Al contrario, un grafo disperso es un grafo con un número de aristas muy bajo, es decir, cercano al que tendría si fuera un grafo vacío.

Como podemos apreciar este, es un grafo bastante disperso ya que tiene pocas aristas respecto al máximo número de aristas posibles, esto es bastante lógico ya que no van a salir todos los pokemons en todos los capítulos y por ende estos no van a estar todos conectados entre sí, para que el grafo fuese completo necesitaría que los 168 nodos tuvieran grado 167, lo cual está bastante lejos.

Estas métricas la considero útil ya que nos indica cuanta interconectividad tiene el grafo ya que un grafo muy interconectado contaría con una densidad muy elevada.

# Graph Density Report | T5

---

## Parameters:

Network Interpretation: undirected

## Results:

Density: 0,240 → Esta métrica se calcula fácilmente haciendo comprobando cuántas aristas tendría el grafo si fuese completo → nodos \* grado máximo nodos/2 =  $214 \times 213 / 2 = 22791$  aristas/aristas máximas →  $5474 / 22 = 0,37$

Esta temporada tiene aún menos densidad que la anterior, esto puede deberse en parte a su mayor número de nodos y su no tan mayor número de aristas, este grafo está más disperso y tiene menos aristas en proporción a los nodos que en la temporada anterior.

# HITS Metric Report | T2

---

## Parameters:

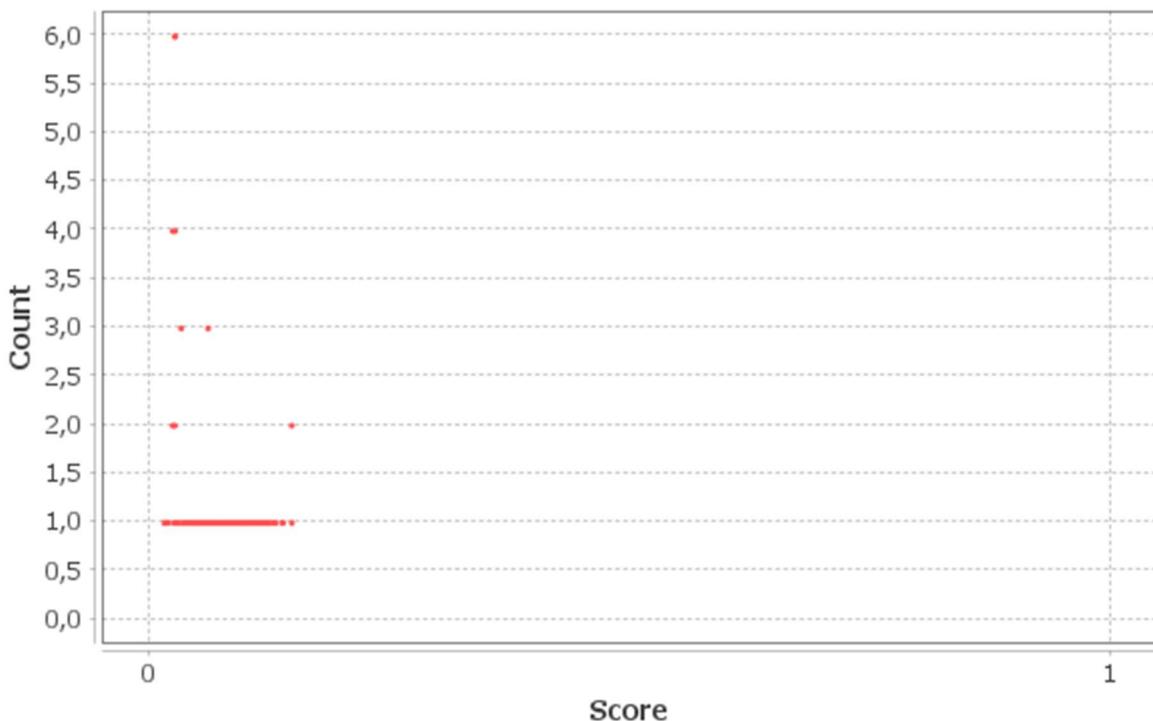
$E = 1.0E-4$

## Results:

La búsqueda de temas inducida por hipervínculos (HITS) (también conocida como centros y autoridades) es un algoritmo de análisis de enlaces que califica las páginas web, desarrollado por Jon Kleinberg. Por lo que no vamos a tener demasiado en cuenta esta métrica debido a que la realidad de nuestro grafo, los tejedores republicanos, se alejan mucho de una página web.

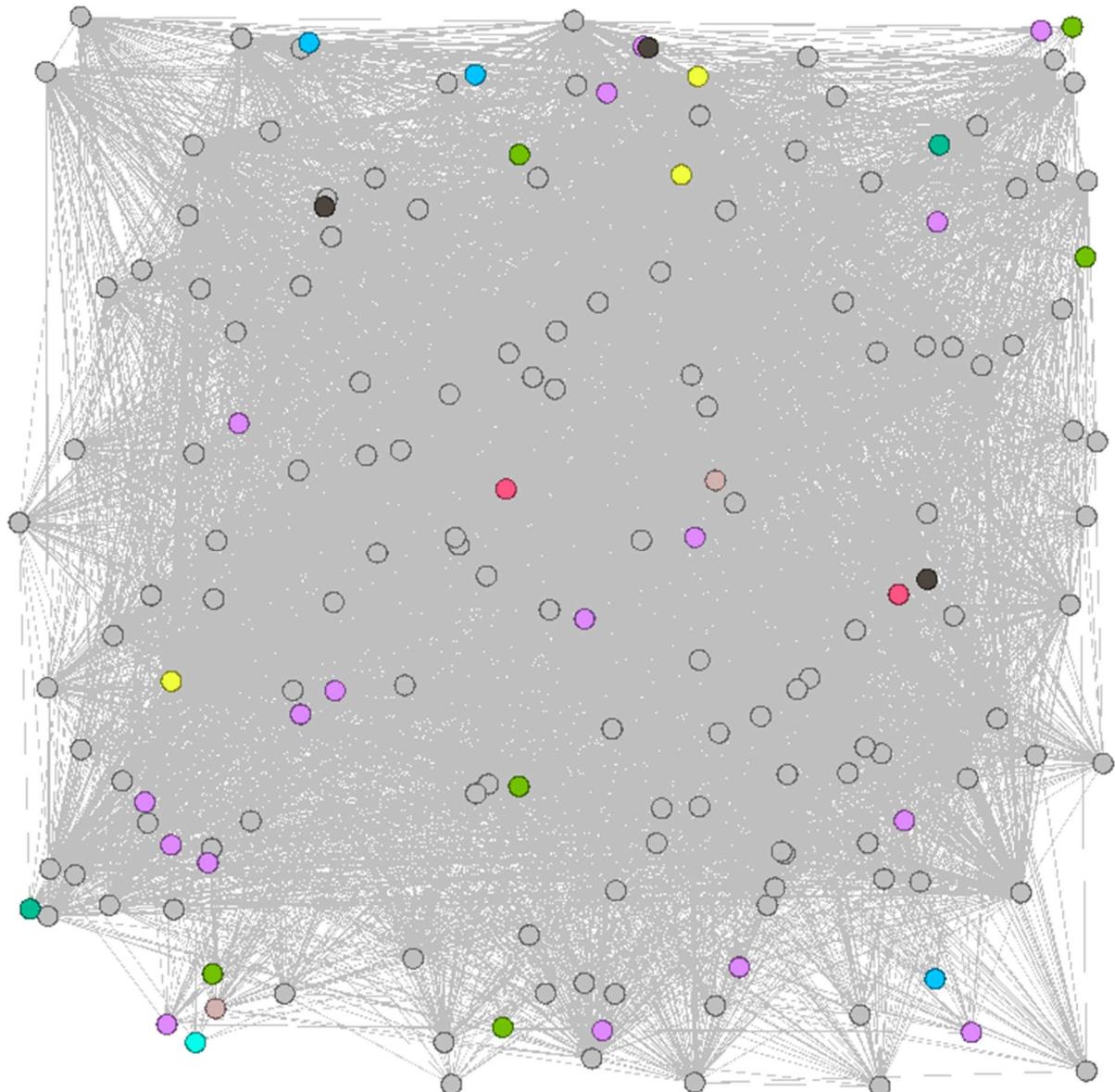
La métrica HITS determina dos valores para una página: su autoridad, que estima el valor del contenido de la página, y su valor central, que estima el valor de sus enlaces a otras páginas. En realidad calcula dos puntajes diferentes: centros y autoridad. La puntuación de autoridad indica el valor de la página (nodo) en sí y los concentradores estiman el valor de los enlaces que salen de la página (nodo). Hits es un algoritmo iterativo en cada iteración: actualice el valor de autoridad de cada nodo para que sea la suma de los valores centrales para cada nodo al que tiene un enlace. Actualice los valores centrales de cada nodo para que sean la suma de los valores de autoridad a los que tiene un enlace. Normalice las puntuaciones de centro y autoridad para todos los nodos normalizando cada valor por la suma del sistema para cada valor. Repita estos pasos (supuestamente hasta que los valores ya no fluctúen).

## Hubs Distribution



Como podrá observar en el gráfico superior los nodos de nuestro grafo tienen un hub realmente bajo no superando el 0,1459.

Tanto en el gráfico de arriba como en la figura que tenemos debajo podemos ver que todos los nodos tienen un hub y authority menor de 0,15, en el grafo están pintados de morado el grupo más numeroso que constan de 0,024 de hub y authority en este grupo encontramos a 17 pokemons, destaca por el otro lado con un hub de 0,1459 los nodos que están pintados de amarillo son los 3 famosos de siempre y el más pequeño de todos está pintado de azul clarito y vuelve a ser el poco sociable “Magneton”.

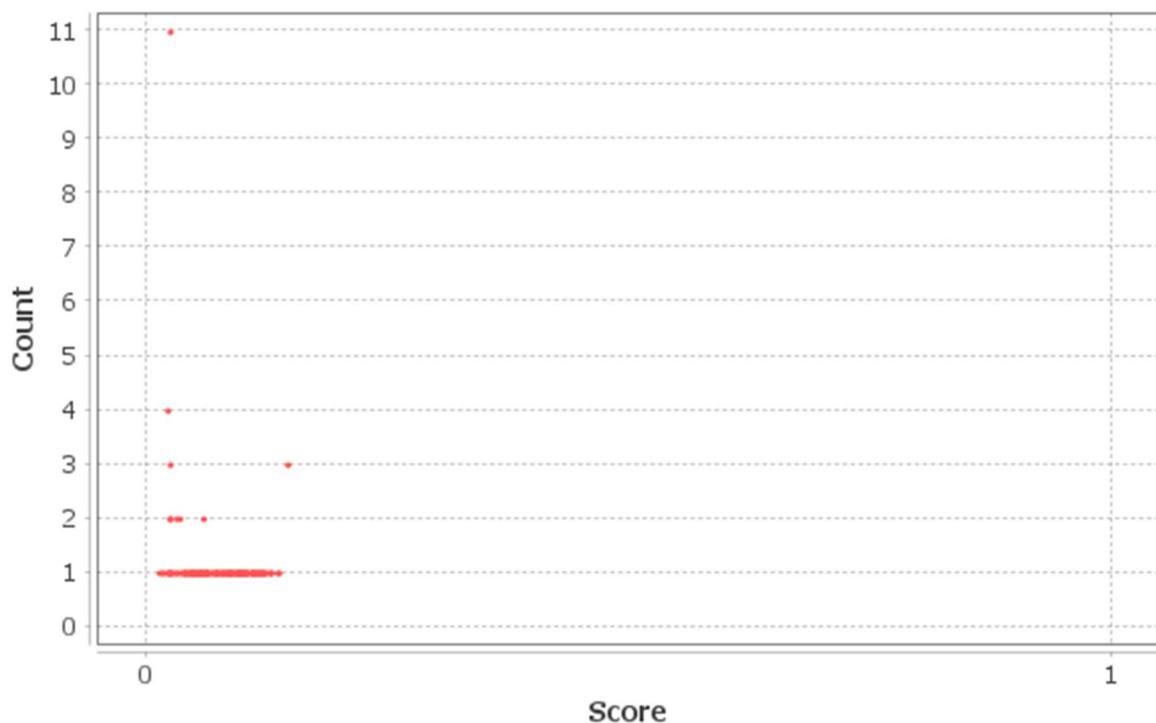


#### Gephi: Partición por hubs o authority

El grafo anterior nos sirve tanto para la hubs distribution como para la Authority distribution ya que en nuestro grafo todos los nodos cuentan con los mismos valores para ambas distribuciones, supongo que esto se debe a que no estamos trabajando con una página web y le faltan parámetros para que cambie. Os dejo una imagen del laboratorio de datos para que corroboréis lo anteriormente mencionado.

Label	Hub	Authority
Pikachu	0.145965	0.145965
Meowth	0.145965	0.145965
Togepi	0.145965	0.145965
Arbok	0.136921	0.136921
Weezing	0.135515	0.135515
Staryu	0.128736	0.128736
Squirtle	0.128432	0.128432
Bulbasaur	0.126674	0.126674
Psyduck	0.12224	0.12224
Vulpix	0.122236	0.122236
Pidgeotto	0.120679	0.120679
Raticate	0.12059	0.12059
Jigglypuff	0.119088	0.119089
Kingler	0.117955	0.117955
Onix	0.117913	0.117913
Tauros	0.117378	0.117378
Krabby	0.11621	0.11621
Mr.Mime	0.115091	0.115091
Poliwag	0.114424	0.114424
Muk	0.113094	0.113095
Venonat	0.112929	0.112929
Starmie	0.111144	0.111144
Lapras	0.110668	0.110668
Oddish	0.110214	0.110214
Voltorb	0.109974	0.109974
Rhydon	0.10937	0.10937
Beedrill	0.108752	0.108752
Mankey	0.108593	0.108593
Chansey	0.107679	0.107679

## Authority Distribution



## **Algorithm:**

Jon M. Kleinberg, *Authoritative Sources in a Hyperlinked Environment*, in Journal of the ACM 46 (5): 604–632 (1999)

# **HITS Metric Report | T5**

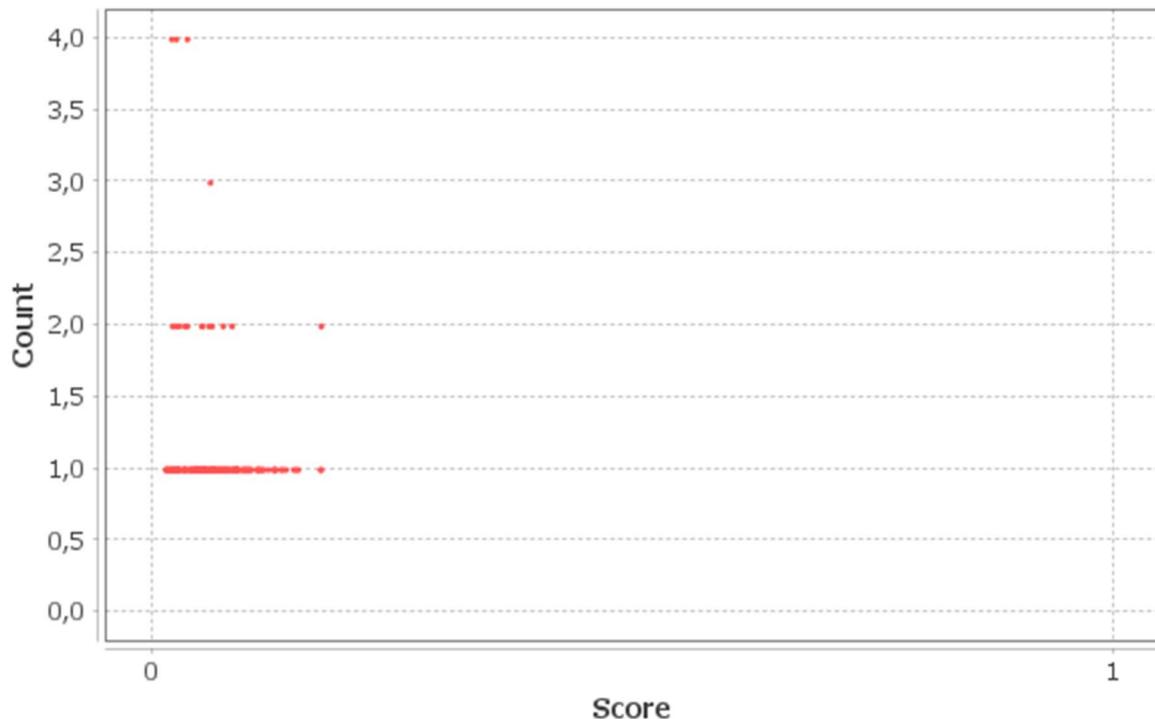
---

## **Parameters:**

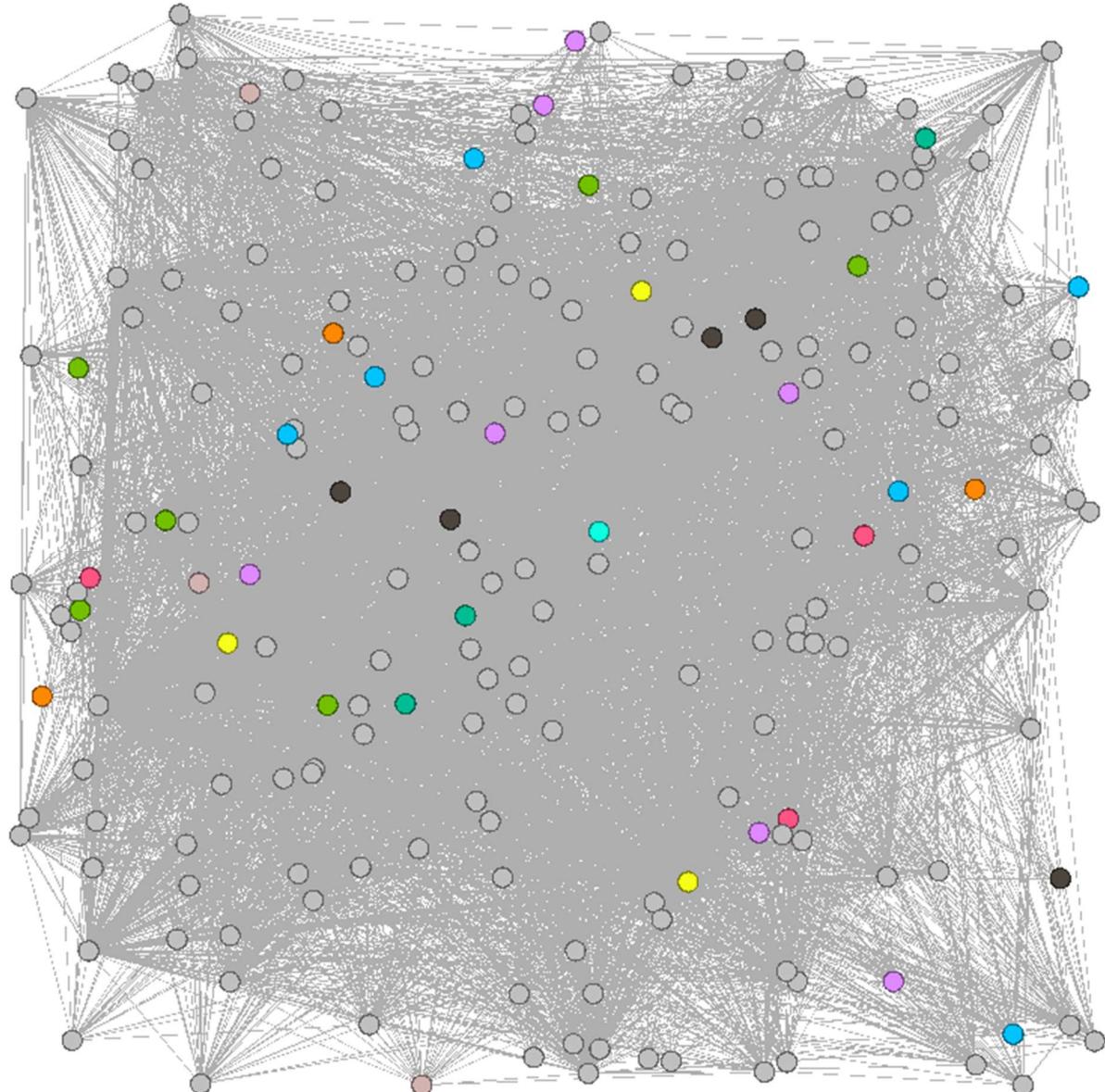
E = 1.0E-4

## **Results:**

**Hubs Distribution**



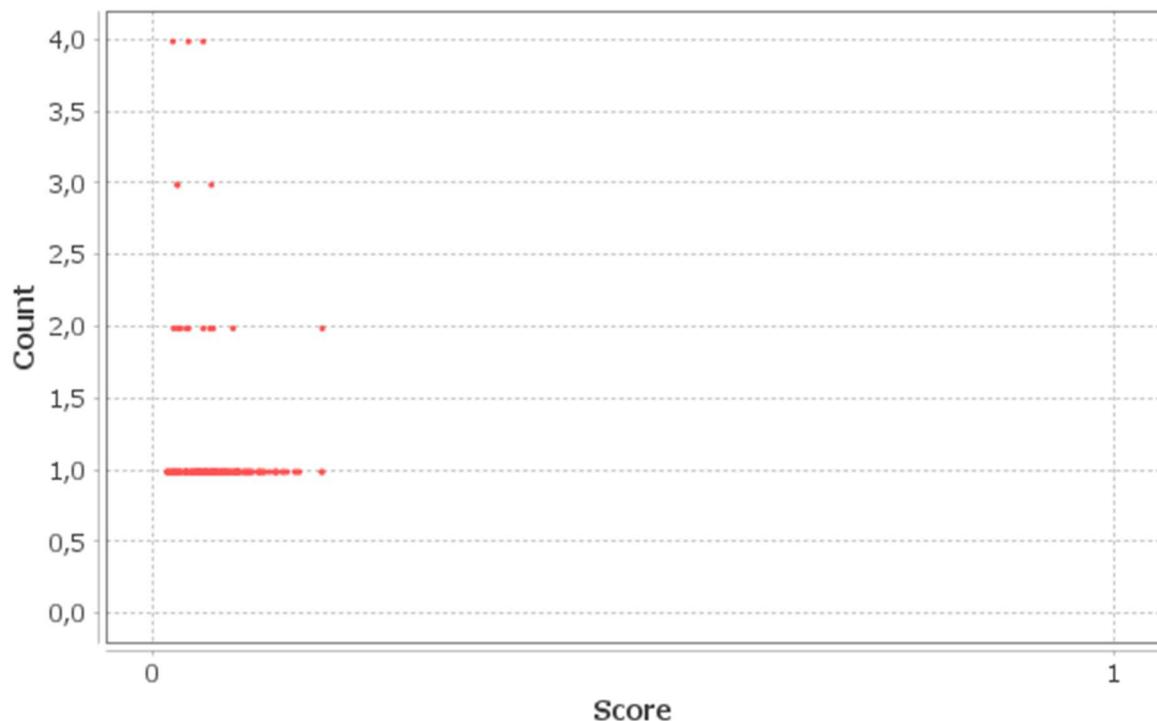
Tanto en el gráfico de arriba como en la figura que tenemos debajo podemos ver que todos los nodos tienen un hub y authority menor de 0,18, en el grafo están pintados de morado el grupo más numeroso que constan de 0,022 de hub y authority en este grupo hay 7 elementos, destaca por el otro lado con un hub de 0,173 los nodos que están pintados de amarillo son los 3 famosos de siempre y el más pequeño de todos está pintado de azul clarito y esta vez el “perdedor” por así decirlo es Unown con un hub y authority 0.011.



Gephi: Partición por hubs o authority

El grafo anterior nos sirve tanto para la hubs distribution como para la Authority distribution ya que en nuestro grafo todos los nodos cuentan con los mismos valores para ambas distribuciones, supongo que esto se debe a que no estamos trabajando con una página web y le faltan parámetros para que cambie. Os dejo una imagen del laboratorio de datos para que corroboréis lo anteriormente mencionado.

## Authority Distribution



Esta métrica al igual que las perteneciente al HITS (Authority distribution y Hubs distribution) están diseñadas para redes sociales y páginas webs, así que no la voy a comentar demasiado, aunque os dejo un grafo en el que los nodos con menor pageRank tienen el mayor tamaño y están pintados de morado, por otro lado el de menor tamaño y pintado de amarillo es el que mayor coeficiente tiene (0,0071) el cual sigue siendo ridículamente pequeño.

## Algorithm:

Jon M. Kleinberg, *Authoritative Sources in a Hyperlinked Environment*, in Journal of the ACM 46 (5): 604–632 (1999)

# PageRank Report | T2

---

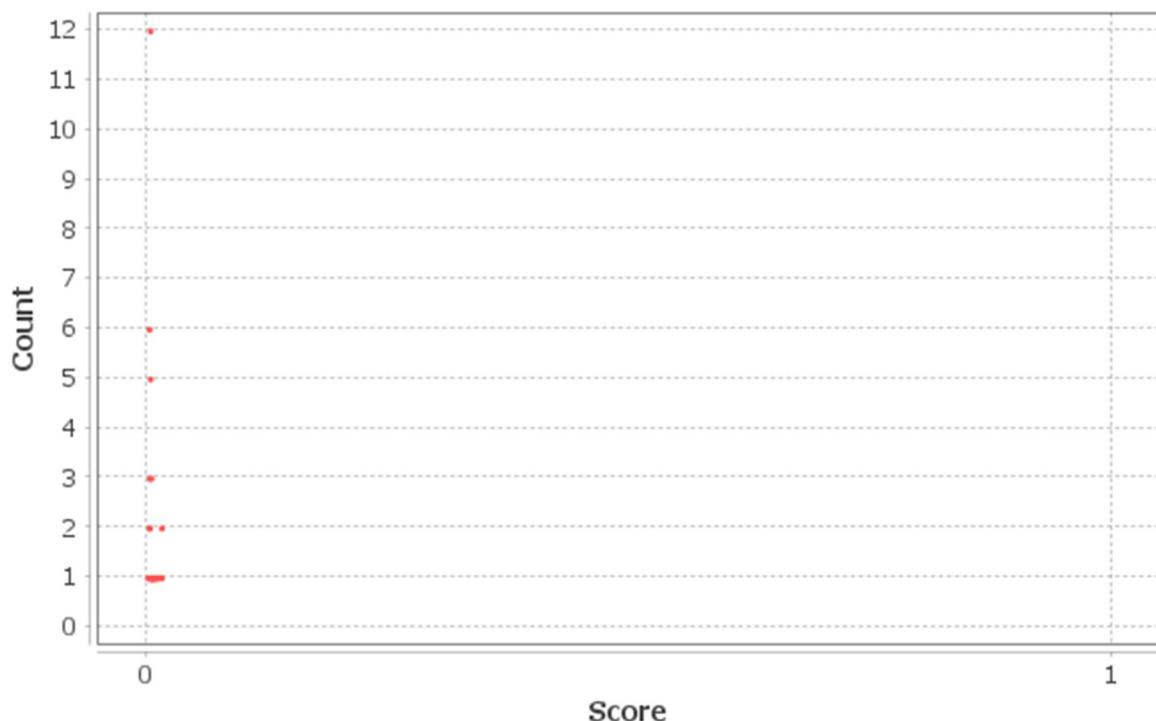
## Parameters:

Epsilon = 0.001

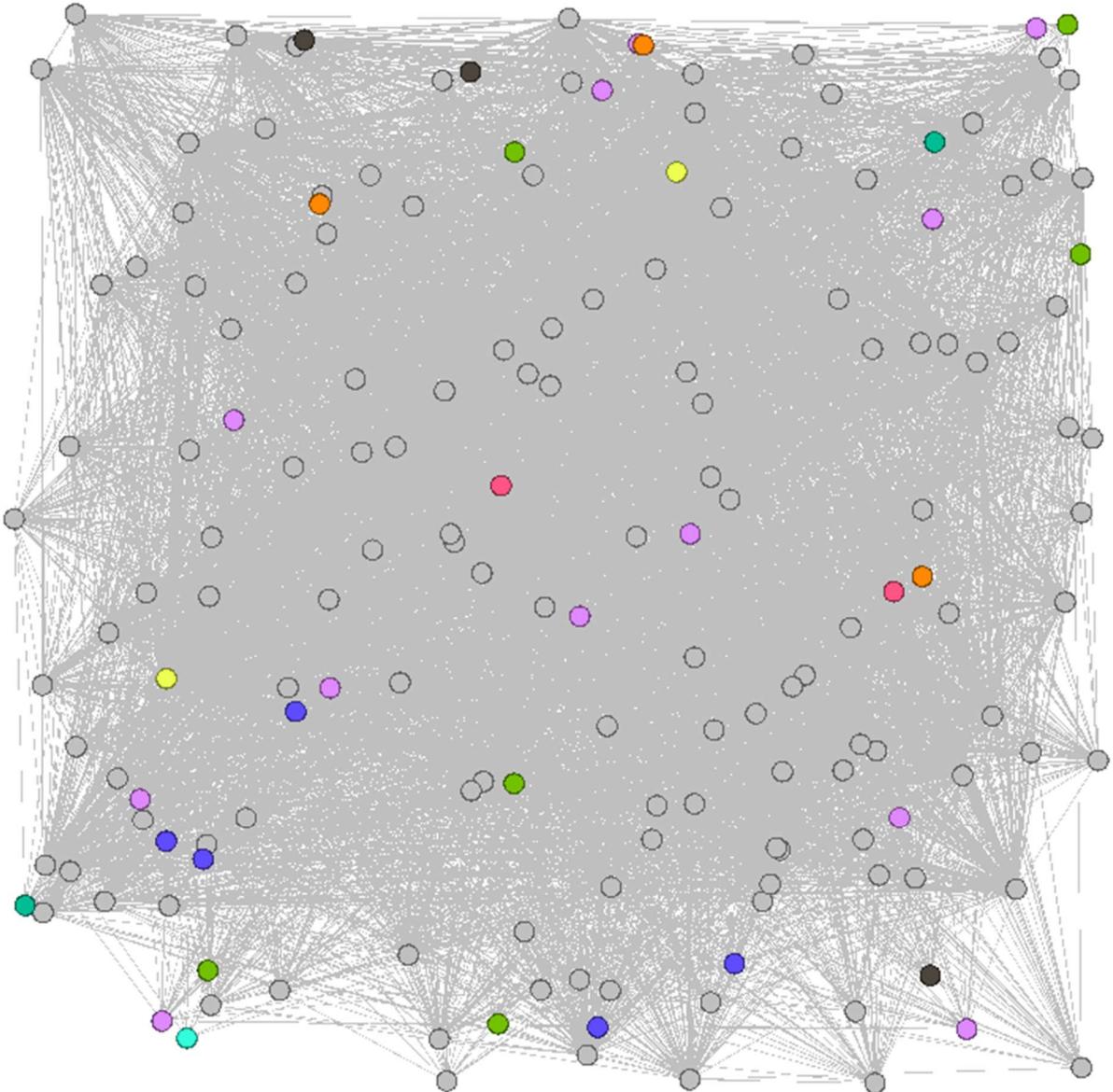
Probability = 0.85

## Results:

**PageRank Distribution**



Esta métrica al igual que las perteneciente al HITS (Authority distribution y Hubs distribution) están diseñadas para redes sociales y páginas webs, así que no la voy a comentar demasiado, aunque os dejo un grafo en el que los nodos con mayor pageRank tienen el mayor tamaño y están pintados de amarillo, por otro lado el de menor tamaño y pintado de azul clarito es el que menor coeficiente tiene (0,0016) el cual sigue siendo ridículamente pequeño. De morado está pintado nuevamente el grupo más numeroso que consta de 12 pokemons y tiene un coeficiente de 0,003.



Esta métrica al igual que las perteneciente al HITS (Authority distribution y Hubs distribution) están diseñadas para redes sociales y páginas webs, así que no la voy a comentar demasiado, aunque os dejo un grafo en el que los nodos con mayor pageRank tienen el mayor tamaño y están pintados de amarillo, por otro lado el de menor tamaño y pintado de azul clarito es el que menor coeficiente tiene (0,0016) el cual sigue siendo ridículamente pequeño. De morado está pintado nuevamente el grupo más numeroso que consta de 12 pokemons y tiene un coeficiente de 0,003.

## Algorithm:

Page, Lawrence and Brin, Sergey and Motwani, Rajeev and Winograd, Terry (1999) *The PageRank Citation Ranking: Bringing Order to the Web*. Technical Report. Stanford InfoLab.

# PageRank Report | T5

---

## Parameters:

Epsilon = 0.001

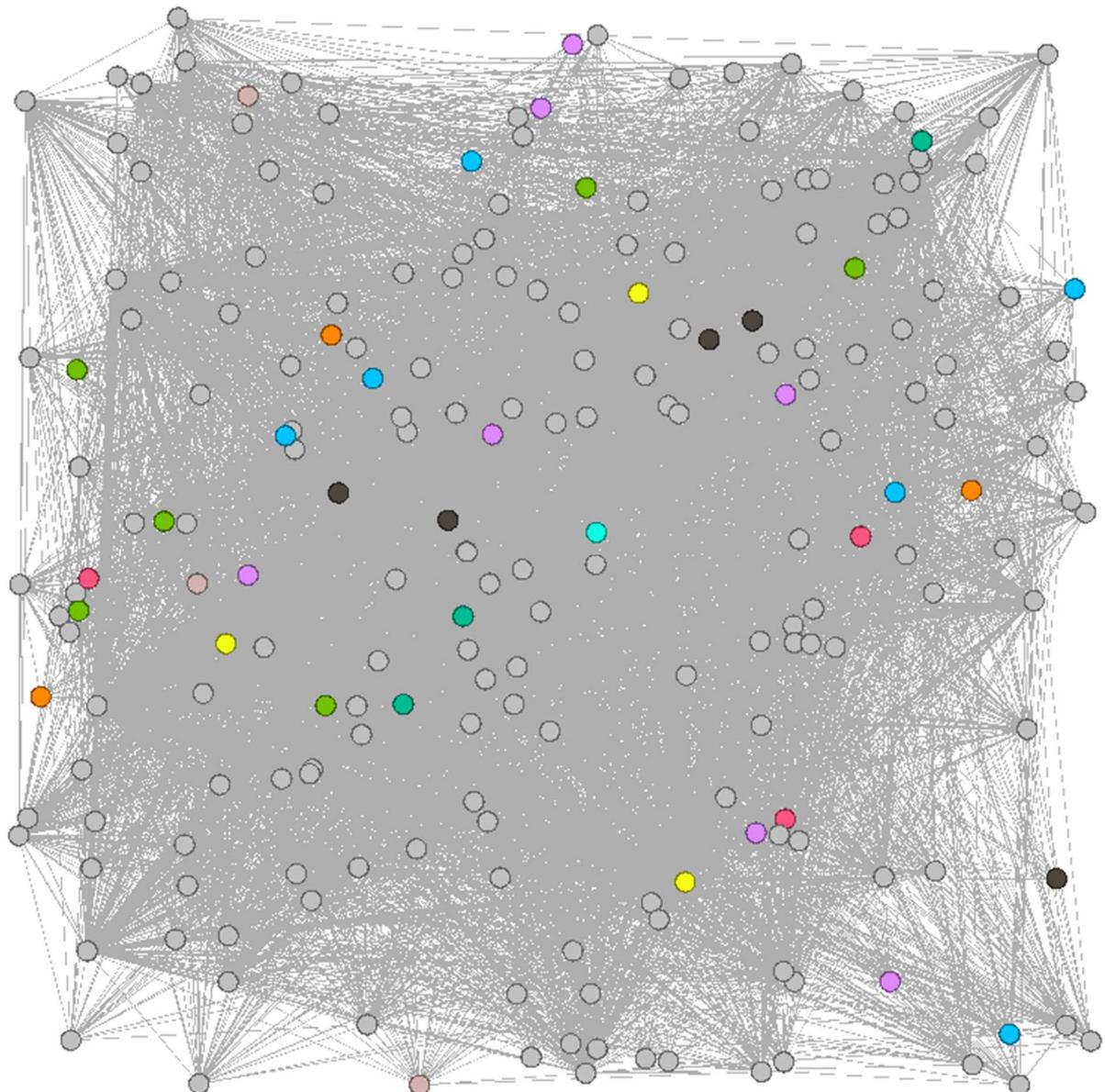
Probability = 0.85

## Results:

**PageRank Distribution**



Os dejo un grafo en el que los nodos con mayor pageRank tienen el mayor tamaño y están pintados de amarillo con un coeficiente de 0.173, por otro lado el de menor tamaño y pintado de azul clarito es el que menor coeficiente tiene (0,0117) el cual sigue siendo ridículamente pequeño. De morado está pintado nuevamente el grupo más numeroso que consta de 7 pokemons y tiene un coeficiente de 0,022.



Gephi: Partición por PageRank

## Algorithm:

Page, Lawrence and Brin, Sergey and Motwani, Rajeev and Winograd, Terry (1999) *The PageRank Citation Ranking: Bringing Order to the Web*. Technical Report. Stanford InfoLab.

# Modularity Report | T2

---

## Parameters:

Randomize: On

Use edge weights: On

Resolution: 1.0

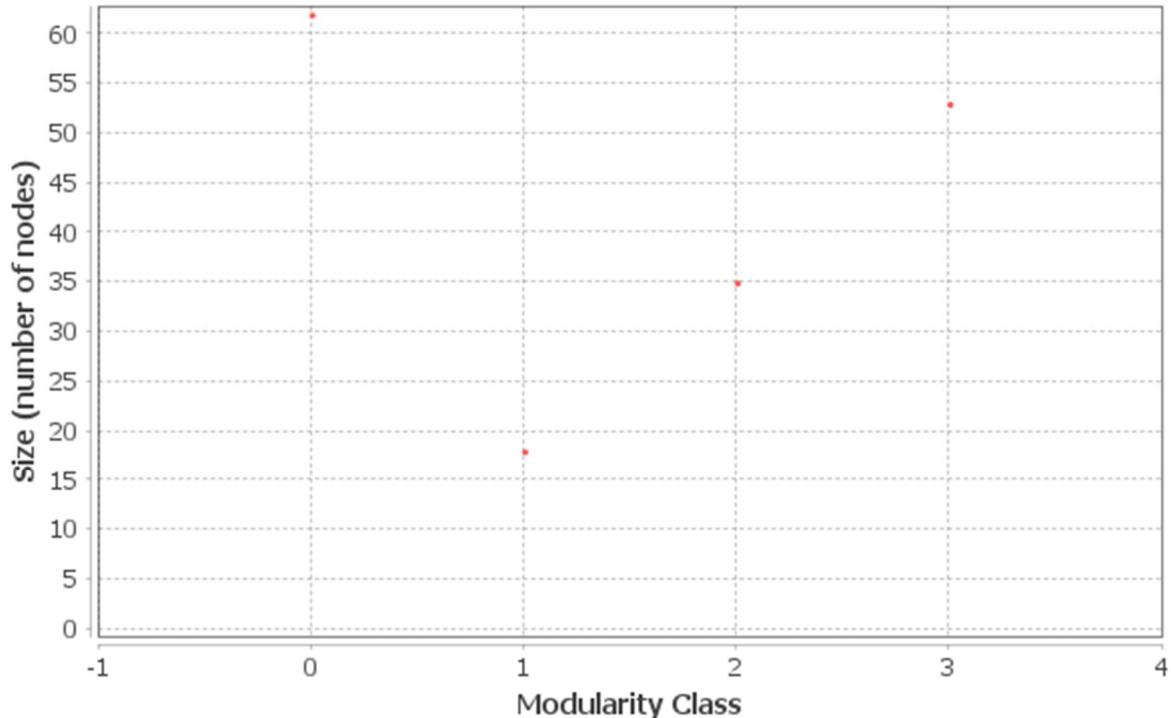
## Results:

Modularity: 0,121

Modularity with resolution: 0,121

Number of Communities: 4

**Size Distribution**



## Algorithm:

Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, Etienne Lefebvre, *Fast unfolding of communities in large networks*, in Journal of Statistical Mechanics: Theory and Experiment 2008 (10), P1000

## Resolution:

R. Lambiotte, J.-C. Delvenne, M. Barahona *Laplacian Dynamics and Multiscale Modular Structure in Networks* 2009

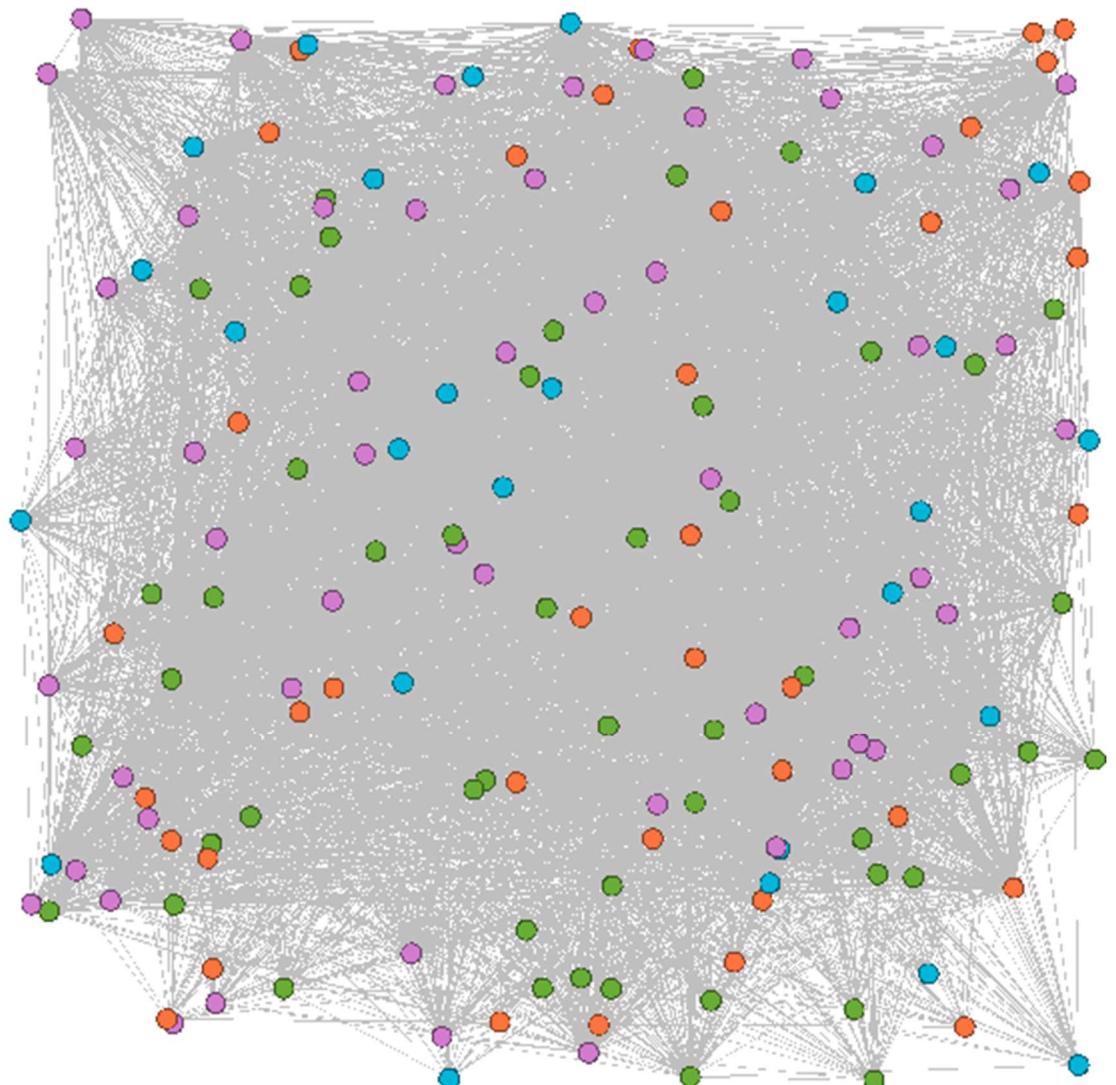
Esta herramienta mide lo bien que puede descomponerse un grafo en comunidades modulares. Una puntuación de modularidad alta indica una estructura interna sofisticada. Esta estructura, a menudo llamada estructura comunitaria, describe cómo la red se comporta en subredes. Se ha demostrado que estas subredes (o comunidades) tienen un significado significativo en el mundo real.

Este algoritmo o métrica para crear nuevos módulos me parece bastante útil ya que si en un estudio quieras dividir los nodos en diferentes comunidades te facilita mucho el trabajo evitándose tener que hacer un análisis intensivo de los atributos de los nodos para luego agruparlos, en resumen, creo que es una herramienta con muchos usos en diferentes campos.

Cada vez que se le da a la métrica para que busque comunidades dentro de la red encuentra comunidades diferentes, de mayor o menor tamaño e incluso varía el número de comunidades que puede encontrar, por lo que puede encontrar características entre los nodos que pueden formar diferentes comunidades dependiendo de cómo los mida.

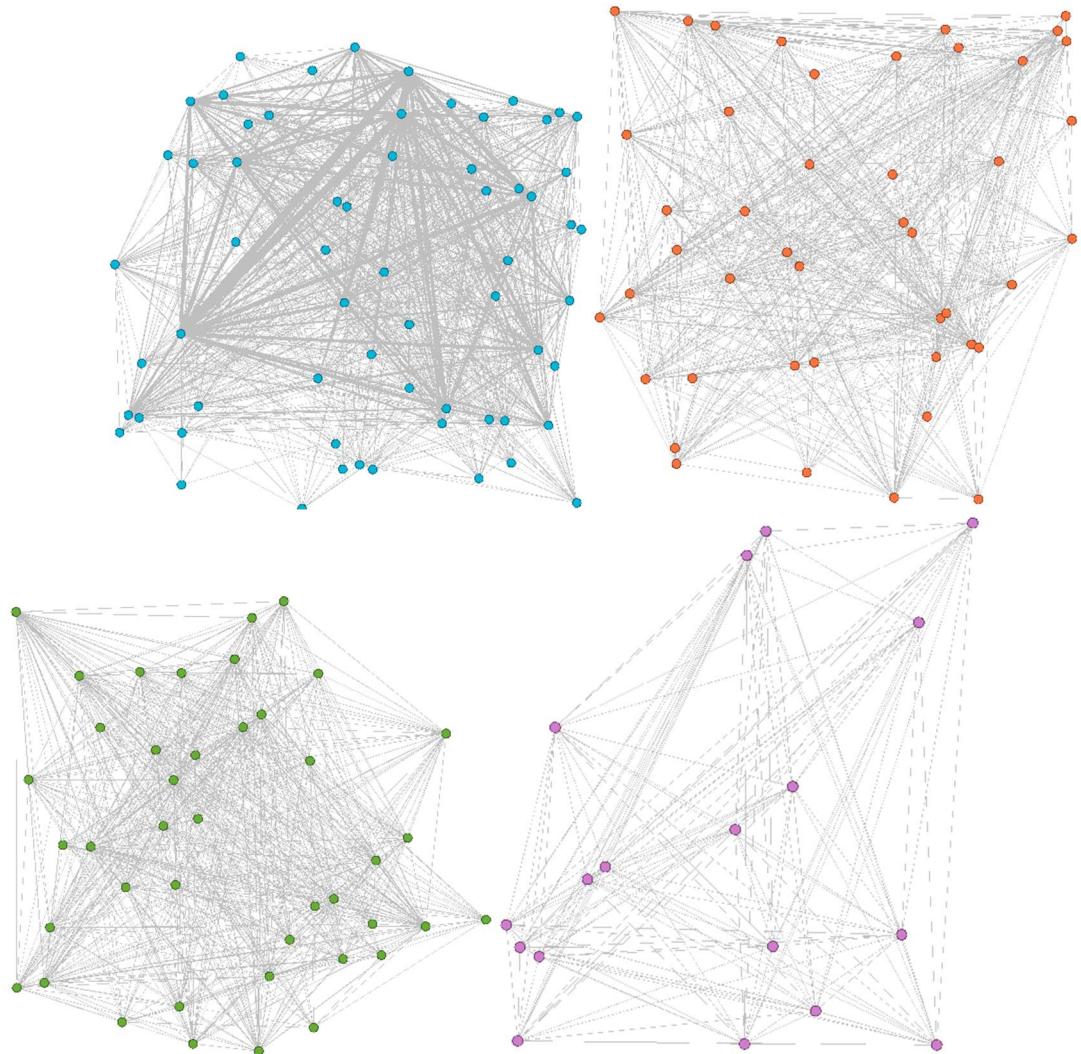
En esta iteración el algoritmo de Modularidad ha dividido la red en 4 comunidades una de ellas predominante (la azul) otras dos casi del mismo tamaño y una muy pequeña como podrán comprobar en las siguientes dos imágenes. He intentado hipotizar de porqué el algoritmo divide de esta forma las comunidades mirando el laboratorio de datos intentando hallar algún patrón que me desvelase el porqué los ha agrupado de esta forma pero después de un buen rato comparando métricas entre nodos de una misma comunidad no he podido hallar en qué se basa el algoritmo para meterlos en la misma comunidad, esto seguramente debido a la complejidad del algoritmo y a mi limitación como humano.

Modularity Class		
3		(37,5 %)
0		(26,79 %)
1		(25 %)
2		(10,71 %)



.Gephi: Partición por Modularity Class

Deabajo les dejo la vista de cada comunidad por separado.



Gephi: Partición por Modularity Class y filtro por partición Modularity Class

## Algorithm:

Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, Etienne Lefebvre, *Fast unfolding of communities in large networks*, in Journal of Statistical Mechanics: Theory and Experiment 2008 (10), P1000

## Resolution:

R. Lambiotte, J.-C. Delvenne, M. Barahona *Laplacian Dynamics and Multiscale Modular Structure in Networks* 2009

# **Modularity Report | T5**

---

## **Parameters:**

*Randomize: On*

*Use edge weights: On*

*Resolution: 1.0*

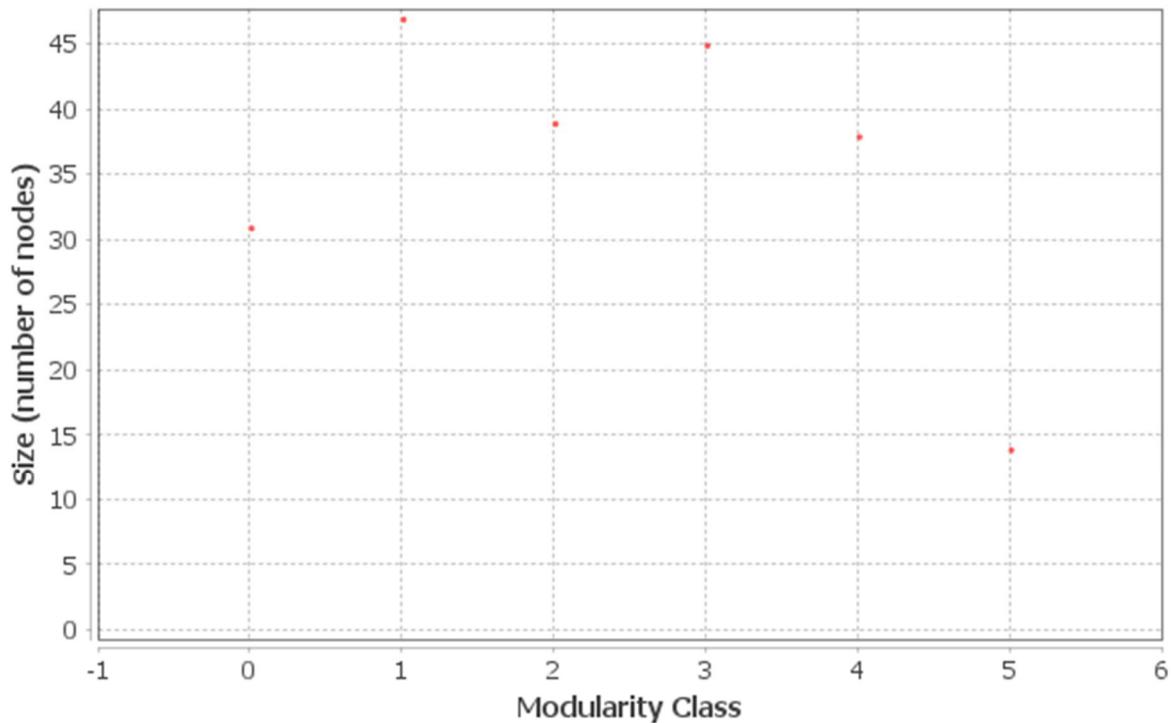
## **Results:**

*Modularity: 0,177*

*Modularity with resolution: 0,177*

*Number of Communities: 6*

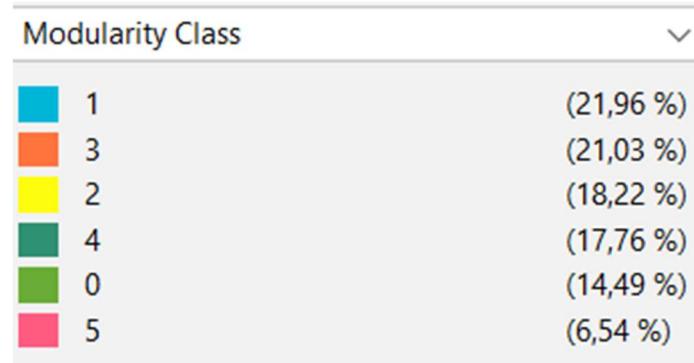
**Size Distribution**



## **Algorithm:**

*Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, Etienne Lefebvre, Fast unfolding of communities in large networks, in Journal of Statistical Mechanics: Theory and Experiment 2008 (10), P1000*

Esta vez el algoritmo ha decidido dividir el grafo en 6 distintas comunidades:

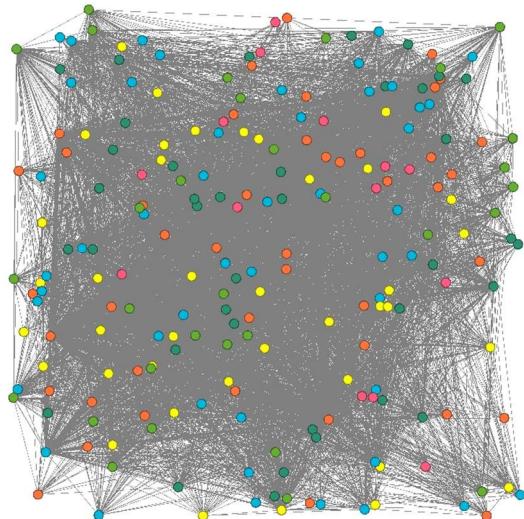


Como podemos observar en la figura superior las comunidades tienen poblaciones bastante parecidas excepto para la comunidad número 5 que tan solo cuenta con un 6,56 del total de los nodos en esta ocasión me he dado cuenta de lo siguiente.

Id	Label	Grad...	Grado	Eccentricity	Closenes...	Harmonic ...	Betweenness C...	Modularity Class
18	Pikachu	1017.0	213	1.0	1.0	1.0	1911.945441	3
38	Meowth	1017.0	213	1.0	1.0	1.0	1911.945441	3
160	Wobbuffet	1007.0	213	1.0	1.0	1.0	1911.945441	3
138	Togepi	1003.0	210	2.0	0.986111	0.992958	1803.364273	3

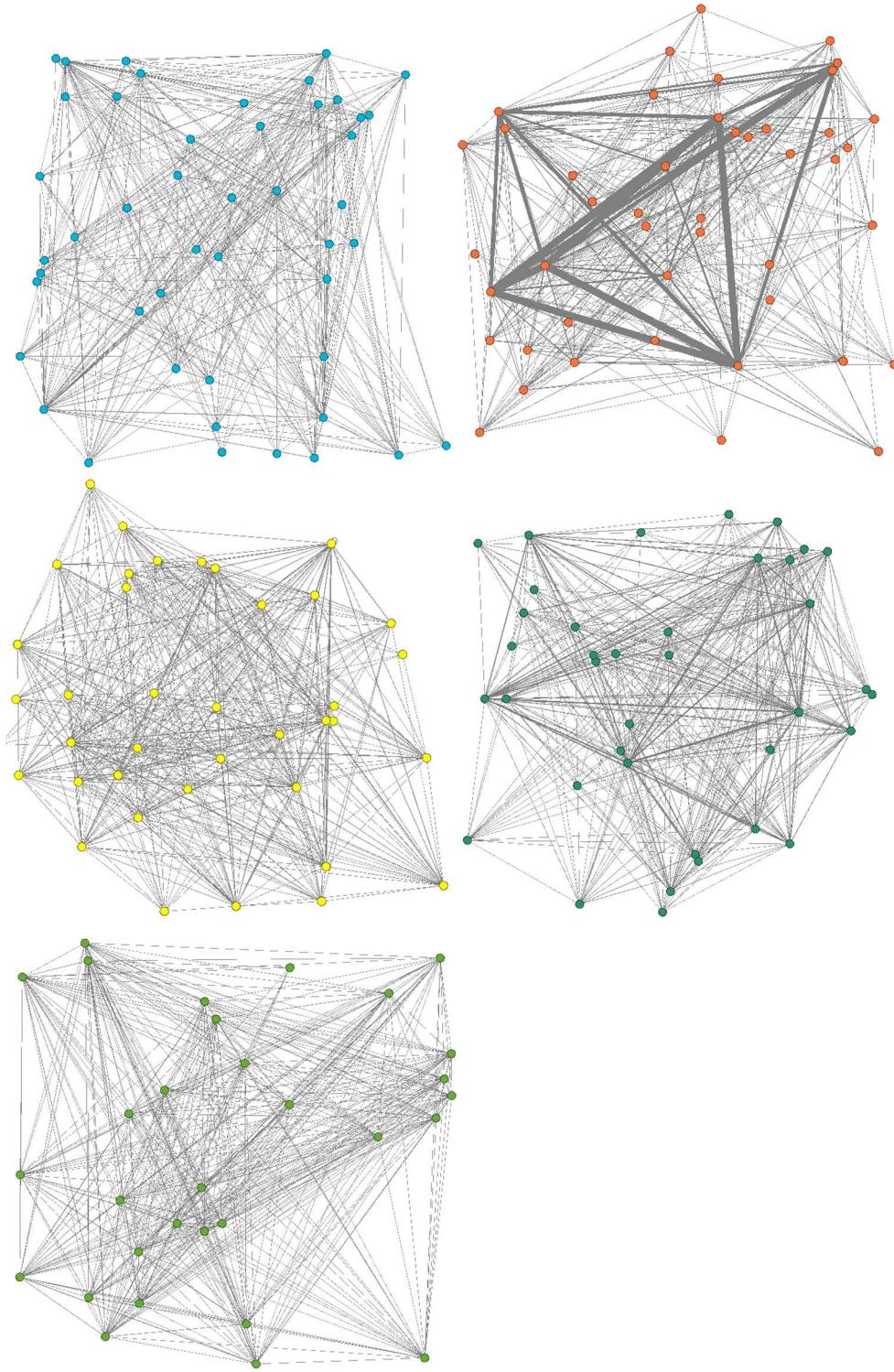
Los pokemons “Pikachu”, “Meowth” y “Wobbuffet” son nodos equivalentes ya que tienen todas sus métricas iguales, en lo único que distan es en sus atributos propios como puede ser su nombre o su tipo, sabiendo esto es asumible que el algoritmo de modularidad los meta en la misma comunidad esto debido a que sus características son muy parecidas, esto se refuerza viendo que “Togepi” también pertenece a su misma comunidad y que esta aunque no tiene los mismos valores que los otros 3 se aproximan bastante.

También es curioso darnos cuenta que en el grupo 5 aunque es el menos numeroso con diferencia teniendo tan solo 14 pokemons (6,54% de la muestra) contiene  $\frac{1}{6}$  de los pokemons legendarios que aparecen en esta temporada, por lo que en términos de poder quizás no está tan alejado a las otras comunidades.



Gephi: Partición por Modularity Class

Aquí debajo le dejo una representación individual de cada comunidad está coloreada de su correspondiente color,



Gephi: Partición por Modularity Class y filtro por partición Modularity Class

## **Resolution:**

*R. Lambiotte, J.-C. Delvenne, M. Barahona Laplacian Dynamics and Multiscale Modular Structure in Networks 2009*

## **Statistical Inference Report /T2**

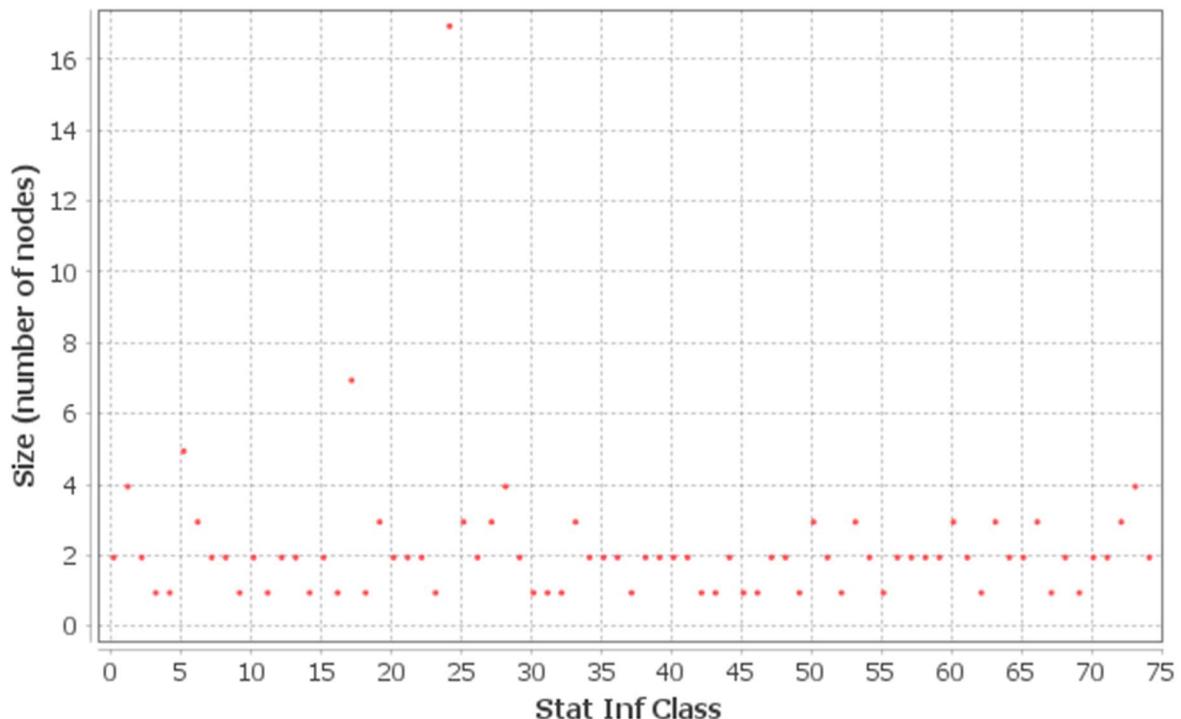
---

### **Results:**

*Description Length: 9595,448*

*Number of Communities: 75*

**Size Distribution**



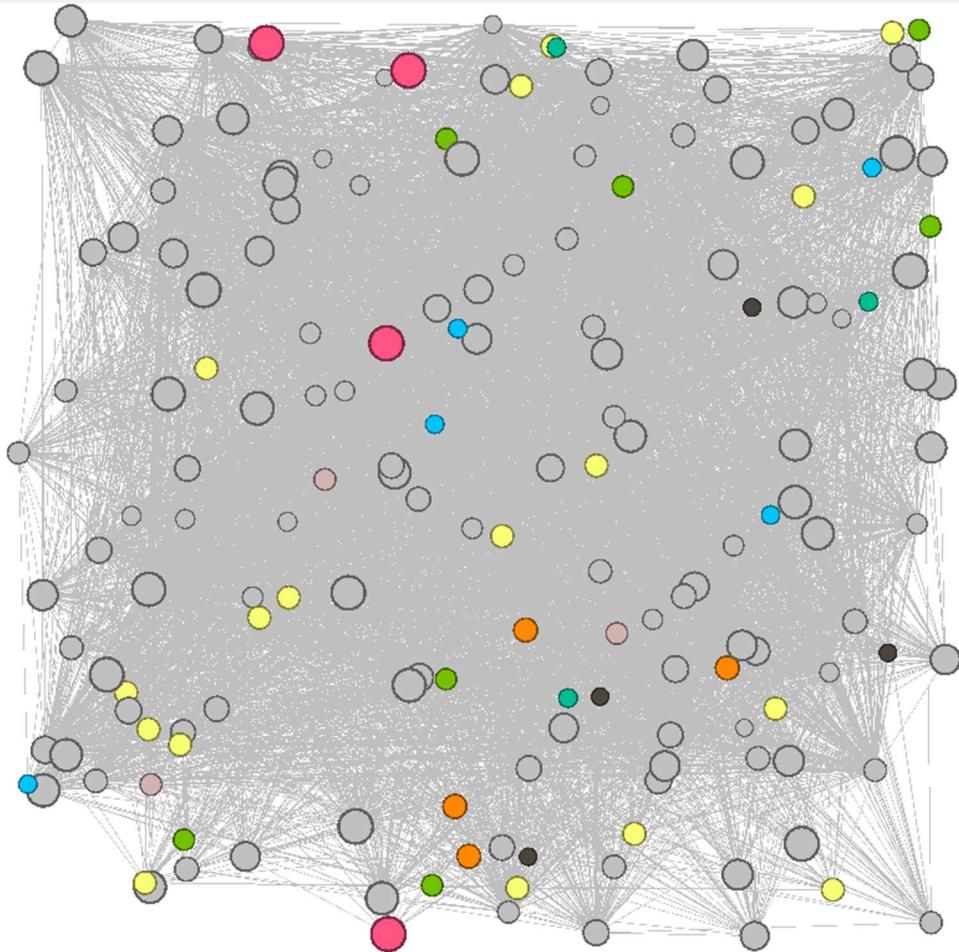
### **Algorithm:**

*Statistical inference of assortative community structures*

*Lizhi Zhang, Tiago P. Peixoto*

*Phys. Rev. Research 2 043271 (2020)*

<https://dx.doi.org/10.1103/PhysRevResearch.2.043271>



Gephi: Partición por InferenceClass

Otro algoritmo para dividir el grafo en comunidades, esta vez ha generado 75 comunidades con la más numerosa pintada de amarillo, es curioso ver que tanto en este algoritmo como en el de modilaridad la temporada 2 que constaba de menos nodos que la 5 los algoritmos hayan generado menos comunidades. El tamaño de los nodos también es dependiendo de la comunidad a la que pertenezcan.

*Bayesian stochastic blockmodeling*

*Tiago P. Peixoto*

*Chapter in “Advances in Network Clustering and Blockmodeling,” edited by*

*P. Doreian, V. Batagelj, A. Ferligoj (Wiley, 2019)*

*<https://dx.doi.org/10.1002/9781119483298.ch11>*

# ***Statistical Inference Report / T5***

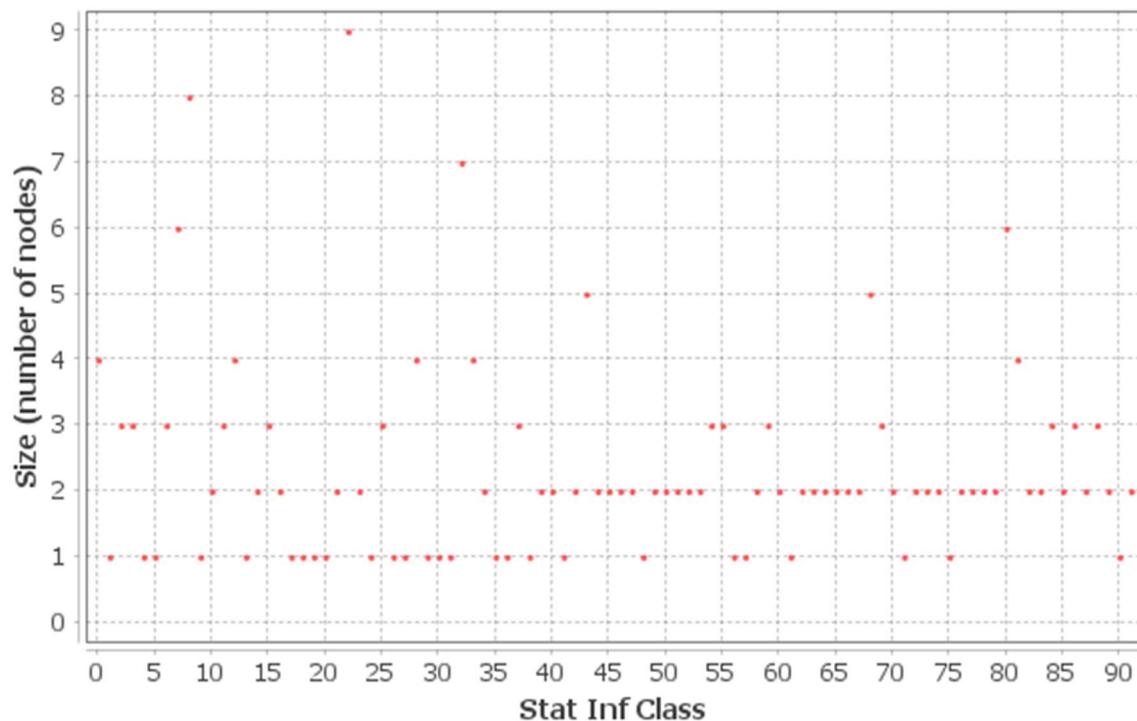
---

## ***Results:***

*Description Length: 12143,528*

*Number of Communities: 92*

### **Size Distribution**



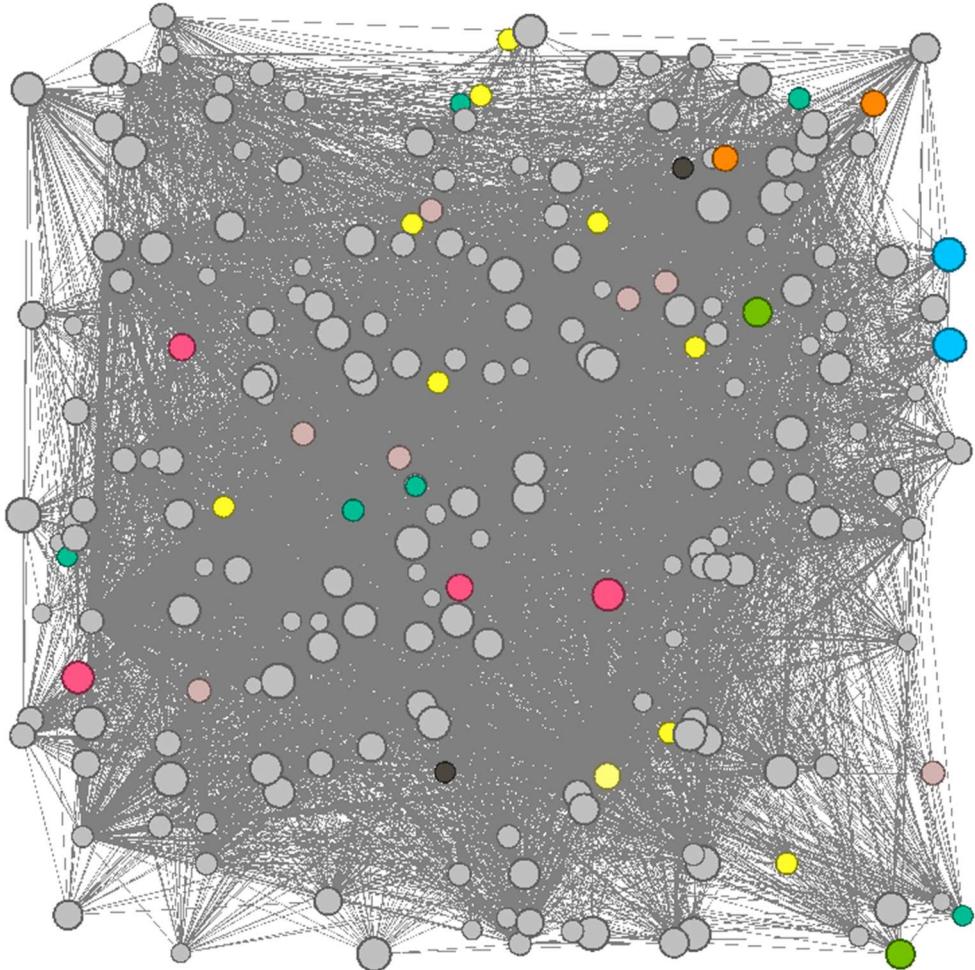
## ***Algorithm:***

*Statistical inference of assortative community structures*

*Lizhi Zhang, Tiago P. Peixoto*

*Phys. Rev. Research 2 043271 (2020)*

Otro algoritmo para dividir el grafo en módulos. Esta vez ha considerado 92 comunidades, divididas como veis en colores y tamaños, las amarillas son el grupo más numeroso, las grises los que menos junto al negro que sólo consta de 2 integrantes (como él hay unos cuantos grupos) también hay “grupos” de un solo individuo. Los tamaños de los nodos están relacionados a la comunidad a la que pertenecen.



Gephi: Partición por InferenceClass

<https://dx.doi.org/10.1103/PhysRevResearch.2.043271>

*Bayesian stochastic blockmodeling*

Tiago P. Peixoto

Chapter in “Advances in Network Clustering and Blockmodeling,” edited by

P. Doreian, V. Batagelj, A. Ferligoj (Wiley, 2019)

<https://dx.doi.org/10.1002/9781119483298.ch11>

# Eigenvector Centrality Report | T2

---

## Parameters:

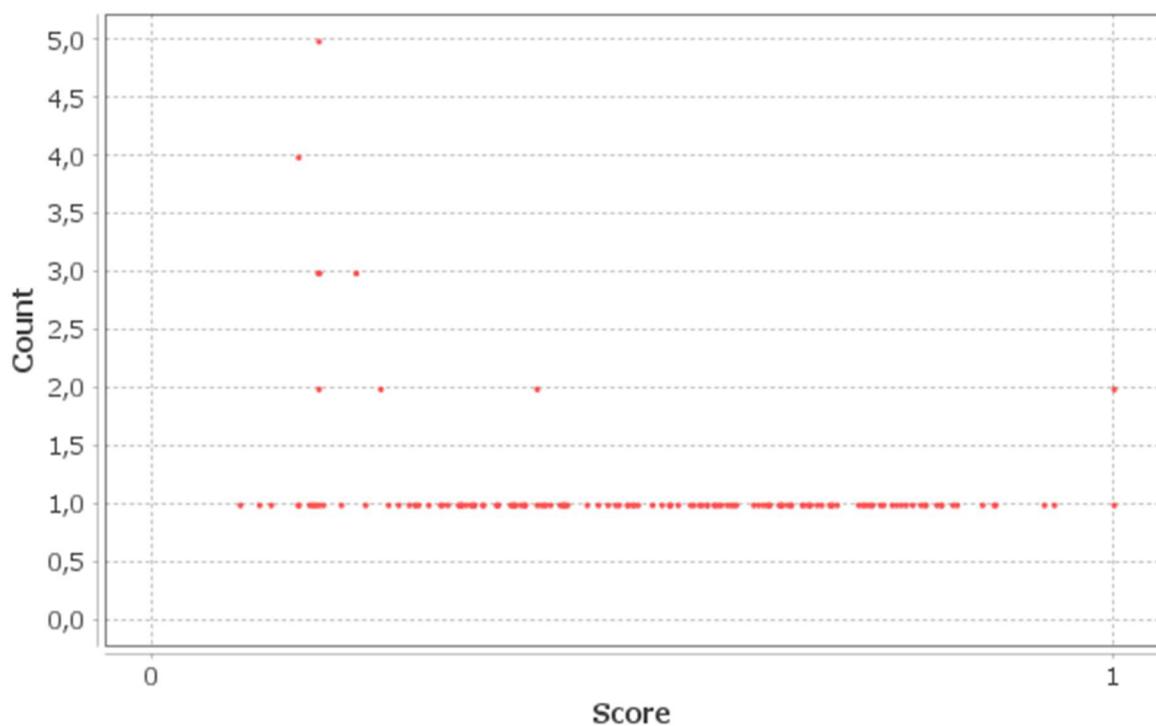
Network Interpretation: undirected

Number of iterations: 100

Sum change: 0.004143020665771935

## Results:

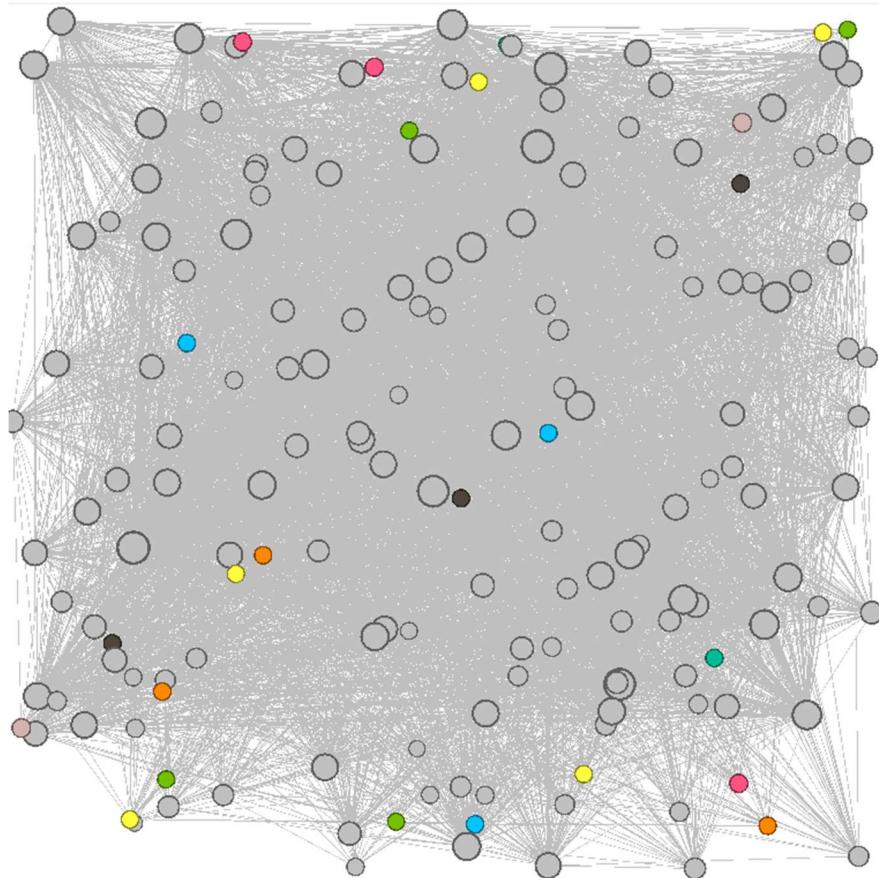
### Eigenvector Centrality Distribution



En análisis de redes sociales, la centralidad de vector propio (en inglés, eigenvector centrality), también llamado prestigio de rango o prestigio de estatus es una medida de centralidad utilizada para cuantificar el nivel de influencia, prestigio o estatus de un nodo o actor en un grafo o red social. Corresponde al principal vector propio de la matriz de adyacencia del grafo analizado. Aquí podemos observar que hay un único nodo que es claramente el más influyente.

En el grafo más abajo podemos observar donde está este nodo más influyente que está pintado de amarillo que casualmente son los nodos “Pikachu”, “Togepi” y “Meowth” estos nodos son los más relevantes en las métricas de HITS sobresaliendo en la distribución de

hubs y la distribución de authority. Esto hace plantearse la importancia de estos nodos, pero personalmente le quito valor porque justo ha sobresalido en las métricas que están diseñadas específicamente para el análisis de redes sociales y de páginas webs, lo cual no es el caso, por lo que esta métrica no la tengo demasiado en cuenta en nuestro análisis. El tamaño de los nodos es según su índice de Eigenvector.



Gephi: Partición por Eigenvector Class

# Eigenvector Centrality Report | T5

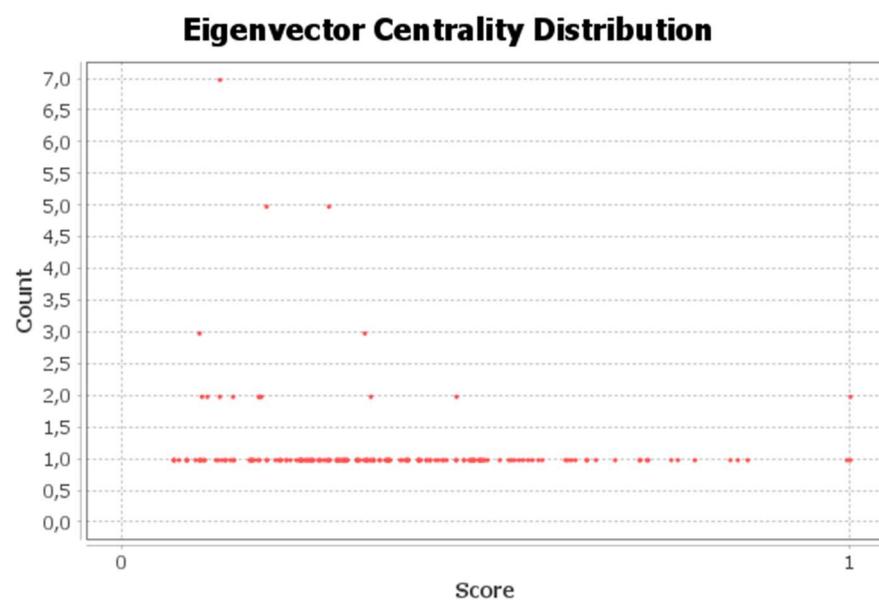
## Parameters:

## Network Interpretation: undirected

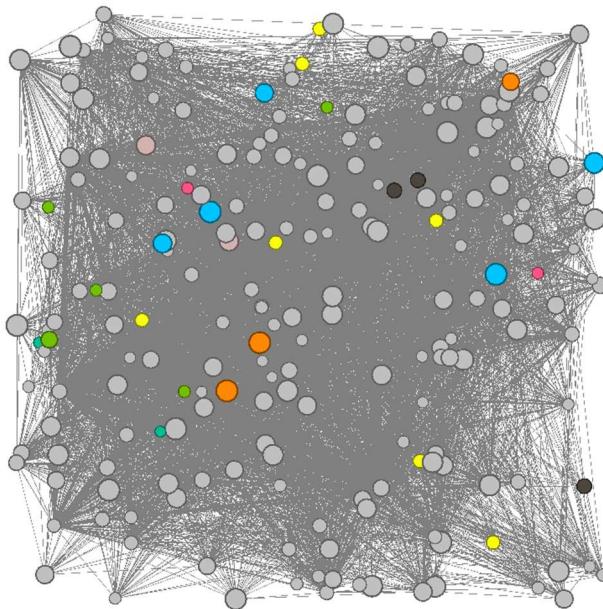
Number of iterations: 100

Sum change: 0.004121402530111479

## Results:



Idem.



Gephi: Partición por Eigenvector Class

## Estudio en Jupyter Notebook

En los cuadernos de jupyter que se entregan están todos los pasos seguidos en el análisis de cada temporada con comentarios sobre el código utilizado.

El primer paso fue cargar la red de cada temporada respectivamente y a continuación obtener las características principales de cada red para su posterior comparación con los modelos de redes aleatorias.

Nº de nodos en el grafo: 168  
 Nº de aristas en el grafo: 5196  
 Tamaño de la componente gigante: 168  
 Número de componentes: 1  
 Camino corto de la componente gigante 1.63  
 Coeficiente de Clustering: 0.78  
 Densidad: 0.3704  
 Diametro: 2

Información del grado  
 -----  
 Grado medio: 61.86  
 Desviación típica: 34.24  
 Grado más alto: 167  
 Grado más bajo: 9

Nº de nodos en el grafo: 214  
 Nº de aristas en el grafo: 5474  
 Tamaño de la componente gigante: 214  
 Número de componentes: 1  
 Camino corto de la componente gigante 1.76  
 Coeficiente de Clustering: 0.76  
 Densidad: 0.2402  
 Diametro: 2

Información del grado  
 -----  
 Grado medio: 51.16  
 Desviación típica: 37.41  
 Grado más alto: 213  
 Grado más bajo: 6

A primera vista observamos que ambas redes tienen una sola componente gigante lo que se podía deducir ya que el pokémon pikachu es el protagonista y aparece en todos los capítulos.

Ambas redes tienen el mismo diámetro y un coeficiente de agrupamiento muy parecido por lo que se entiende que los pokémon que aparecen coinciden con otros pokémon de mucha popularidad ya que ambas redes tienen nodos con un grado elevado.

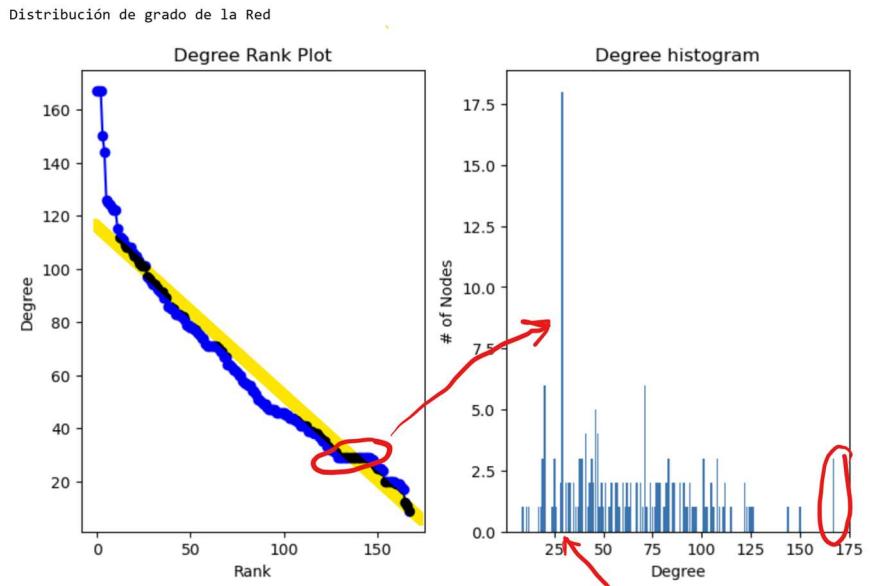
Sin embargo, estas distan en el grado medio y levemente en el camino medio. Además de tener más nodos, la quinta temporada tiene un grado medio más bajo y en consecuencia una densidad de red considerablemente más baja. Esto también se nota en que el camino mínimo medio del grafo de la red de la quinta temporada es mayor que el de la segunda. No obstante el grado más alto lo encontramos en la quinta temporada ya que hay pokémon que aparecen en todos los capítulos y en esta temporada hay más episodios.

Pintamos la distribución de grado para tener una mejor idea del grado de los nodos de la red globalmente. Observamos que hay 3 nodos con un grado muy alto (168 y 213) que son los pokémon que aparecen en todos los capítulos de las temporadas.

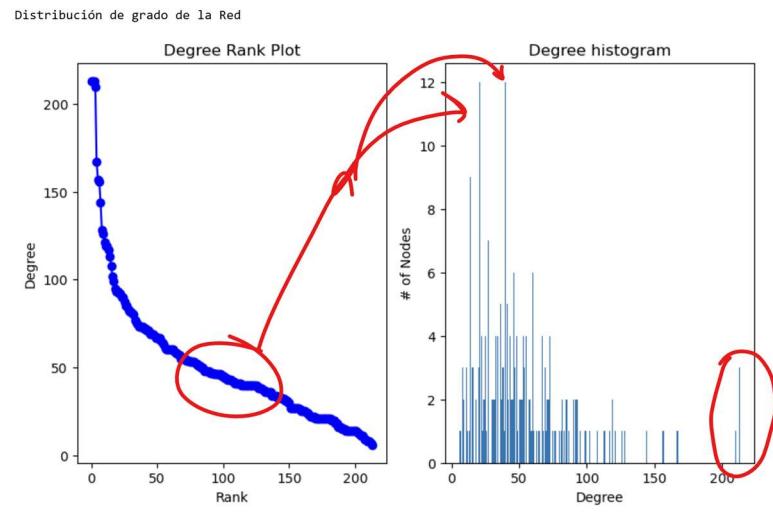
## Segunda Temporada

También se puede diferenciar una zona más poblada rondando el grado 30 de lo que podemos deducir que pertenece a los pokémon más estándar en cuanto a protagonismo que aparecen un número menor de capítulos respecto al valor del grado medio. Esta peculiar acumulación se puede entender como una proporción de capítulos cómoda en la que los creadores confiaban para destacar la cantidad de distintos pokémon que hay pero sin solapar la historia de los pokémon con mayor peso argumental.

Por otra parte, se destaca con una línea amarilla la semejanza con una distribución lineal en la que salvo en los extremos la densidad de nodos es bastante homogénea dentro de una notable respecto del grado medio. Esto difiere con el modelo teórico de una red aleatoria



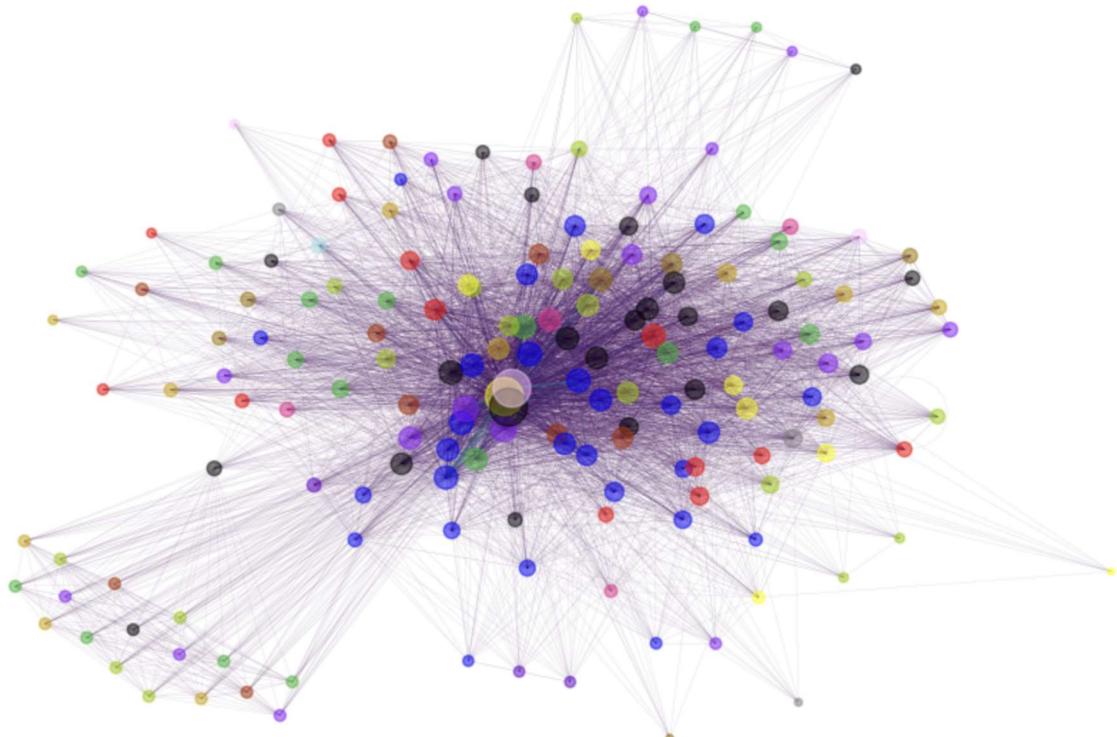
## Quinta Temporada



En este caso también hay zonas más densas pero no tan notablemente. En general hay menos homogeneidad en la distribución de grado y se nota que la mayoría de nodos tiene un grado menor al grado medio de la red.

A continuación, dibujamos ambas redes diferenciando los nodos por color según su tipo y su tamaño dependiendo del grado. El grosor de las aristas depende de su peso.

## Segunda Temporada

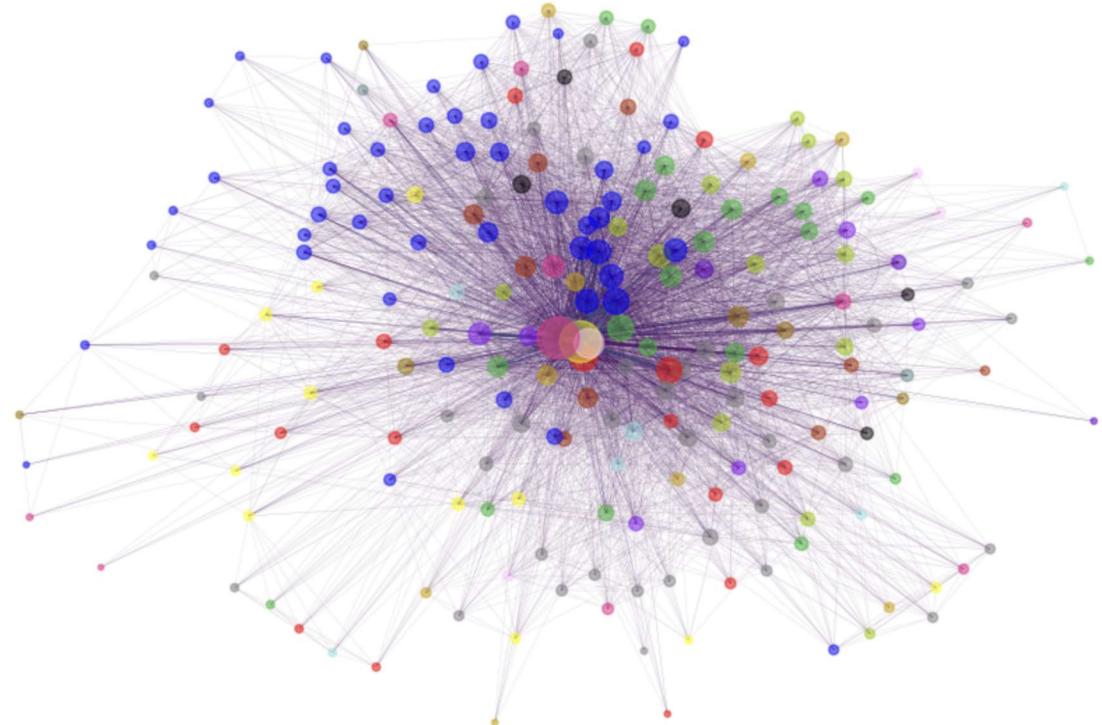


Observamos como aparecen los 3 pokémon que salen en todos los episodios en el centro de la red con el mayor tamaño. Pikachu amarillo (Tipo eléctrico), Togepi rosa (Tipo Hada) y Meowth negro (Tipo normal). Además se puede observar que el tipo de pokémon más frecuente dentro de un tamaño mediano/grande son el agua (azul) y el normal (negro). Por lo que se deduce que son los pokémon más habituales en la región del mundo pokémon en la que ocurre la trama. Se entiende que es una zona con mucha naturaleza en la que hay lagos y ríos, así como mar al tratarse de una isla.

También podemos ver pokémon de tipo veneno (morado) con un tamaño considerable y céntricos en la red pero no son muchos nodos. Estos nodos seguramente representen los pokémon del Team Rocket que la mayoría son de este tipo y los combates en la serie con este equipo son frecuentes.

Por otra parte, observamos una zona alejada en la que predominan los pokémon de tipo bicho. Esto se puede interpretar como que en algunos capítulos concretos, ya que el grosor de las aristas es bajo, lo protagonistas van a una zona forestal en la que hay muchos pokémon de este tipo y los encuentran al mismo tiempo. Por lo que en la red aparecen ligados entre sí pero sin tener gran peso respecto al resto de nodos.

## Quinta Temporada



En este caso vemos como también hay 3 pokémon protagonistas, en este caso Pikachu, Wobbuffet y Meowth. En esta temporada también hay un dominio de apariciones por parte de los pokémon de tipo agua que es el tipo con más número de pokémon distintos apareciendo

en la serie. En esta temporada se puede apreciar un mayor protagonismo de los pokémon de tipo planta.

Es curioso como en esta temporada se le da protagonismo a Wobbuffet un pokémon de tipo psíquico el cual es poco frecuente en las temporadas analizadas. Este pokémon pese a no ser destacado en sus características de combate en los videojuegos adquiere un protagonismo memorable en la serie siendo el compañero inseparable del Team Rocket en tantos malficios.

En ambas temporadas se aprecia que hay ciertos tipos menos frecuentes como lo son el hielo, dragón, hada, tierra o roca y en concreto los tipos siniestro y acero que en la segunda temporada no aparecen.

## Comparación con el Modelo de Configuración

Este modelo se genera a partir de una secuencia de grados que en este caso generamos aleatoriamente ajustando los parámetros para obtener unos valores que generen una red parecida a cada una de las que modelan las temporadas:

### Segunda Temporada

```
valores_aleatorios = np.random.normal(83, 34, 168)
```

RED 1

```
Número de nodos: 168.0
Número de aristas: 5134.84
Camino mínimo medio: 1.6
Grado medio: 61.13
Radio medio de la red 2.0
Diámetro medio de la red 2.9
Número de componentes conexas: 4
Media del coeficiente de clustering 0.43789
Coeficiente de clustering global 0.44208
```

### Quinta Temporada

```
valores_aleatorios = np.random.normal(65, 26, 214)
```

RED 1

```
Número de nodos: 214.0
Número de aristas: 5481.36
Camino mínimo medio: 1.8
Grado medio: 51.23
Radio medio de la red 2.0
Diámetro medio de la red 3.1
Número de componentes conexas: 2
Media del coeficiente de clustering 0.29343
Coeficiente de clustering global 0.29425
```

Analizando los resultados obtenidos se puede concluir que este modelo genera redes con valores similares para muchas de las métricas pero no nos asegura la componente gigante y el clustering es notablemente bajo. En general las métricas varían considerablemente de una ejecución a otra, por lo que este método no parece del todo fiable.

## Comparación con el Modelo Erdos-Renyi

Para la comparación de este modelo tomamos una probabilidad obtenida a partir de las siguientes fórmulas teóricas sustituyendo los datos de nuestra red en las variables correspondientes:

$$N_p = \binom{N}{2} = \frac{N(N-1)}{2}$$

$$p_c = \frac{M}{N_p} = \frac{2M}{N(N-1)}$$

En nuestro caso  $p_c = 0,368$  y  $0,240$ .

Además sabemos que para este modelo el coeficiente de clustering se rige por la siguiente fórmula:

### Clustering coefficient of a random network

$$C_i = \frac{2\langle L_i \rangle}{k_i(k_i - 1)} = p = \frac{\langle k \rangle}{N}.$$

Por lo que el valor esperado del coeficiente de clustering para la red generada en este modelo es de  $0,368$  y  $0,240$ . Los resultados obtenidos con networkx son los siguientes:

#### Segunda Temporada

RED 1

---

Número de nodos: 168.0  
 Número de aristas: 5174.92  
 Camino mínimo medio: 1.6  
 Grado medio: 61.61  
 Radio medio de la red 2.0  
 Diámetro medio de la red 2.0  
 Número de componentes conexas: 1  
 Media del coeficiente de clustering 0.36944  
 Coeficiente de clustering global 0.3693  
 Densidad 0.3689

#### Quinta Temporada

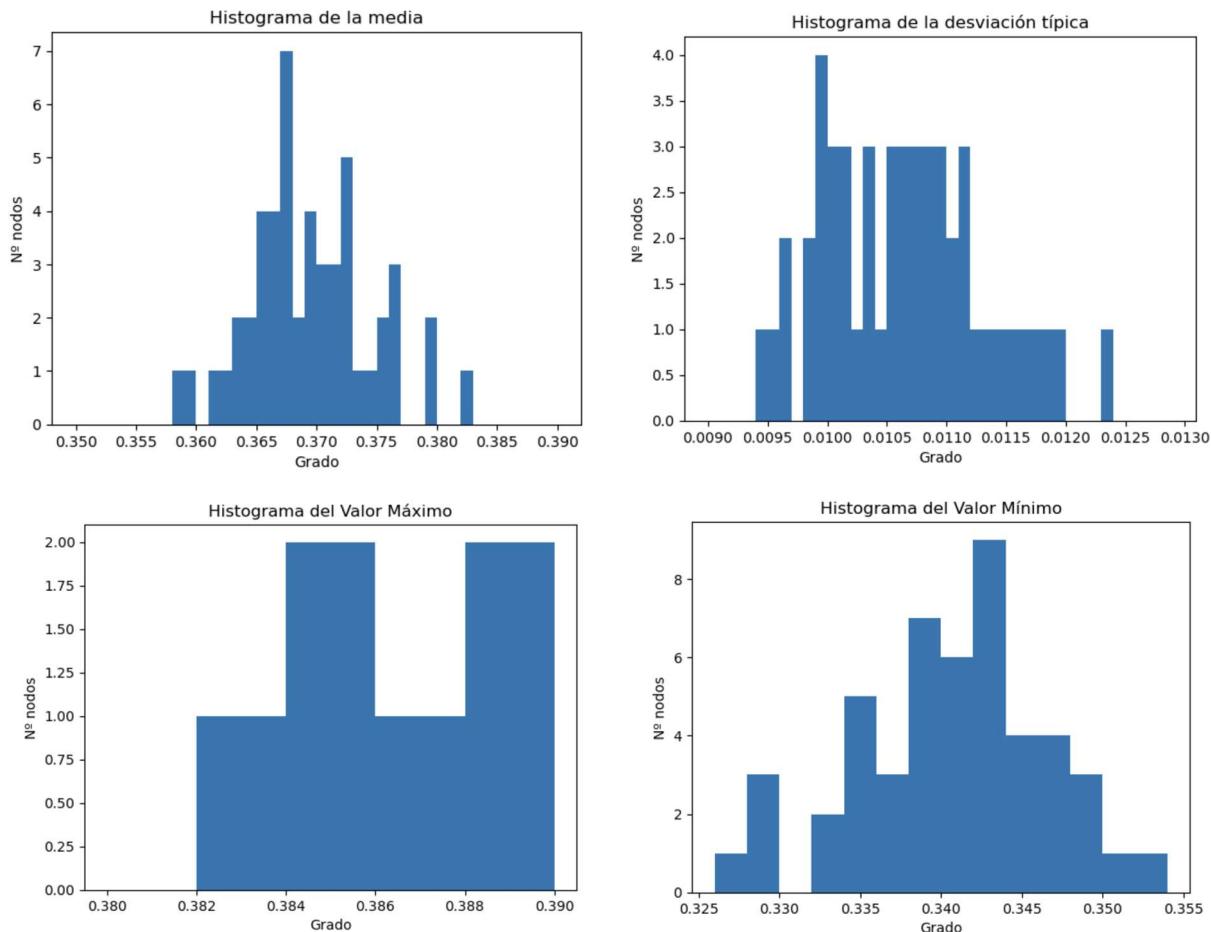
RED 1

---

Número de nodos: 214.0  
 Número de aristas: 5470.78  
 Camino mínimo medio: 1.8  
 Grado medio: 51.13  
 Radio medio de la red 2.0  
 Diámetro medio de la red 2.1  
 Número de componentes conexas: 1  
 Media del coeficiente de clustering 0.23991  
 Coeficiente de clustering global 0.23982  
 Densidad 0.24004

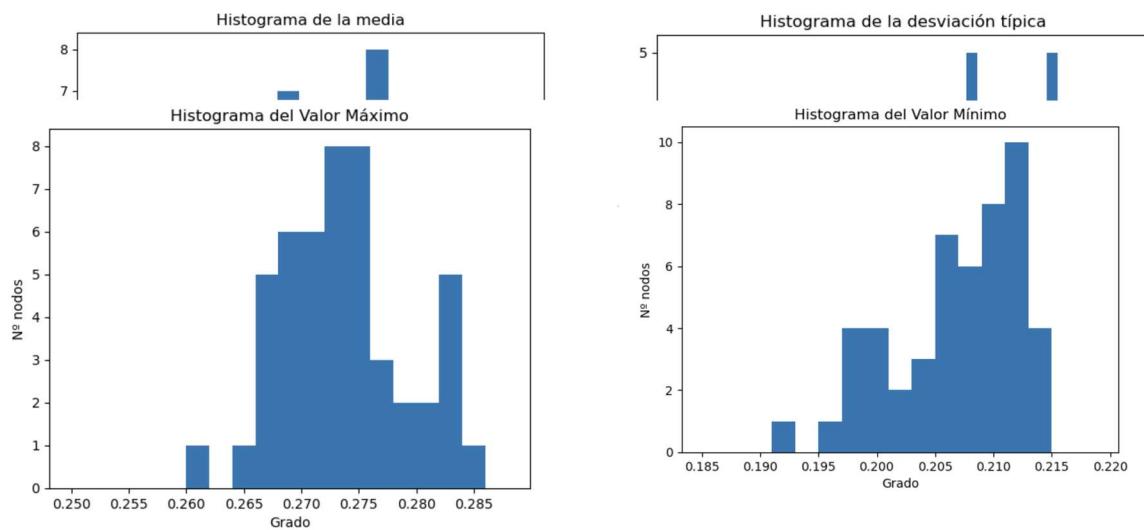
Histogramas de la media, desviación estándar, valor máximo y valor mínimo del coeficiente de clustering de las 50 redes generadas:

## Segunda Temporada



## Quinta Temporada

Efectivamente se confirma la hipótesis previa con el coeficiente de clustering obtenido a partir de la generación de redes con este modelo. Salvo este, el resto de propiedades de las redes generadas son muy aproximadas al de nuestra red real. Esto nos lleva a pensar que el modelo ER es bastante eficaz para generar redes con un grado medio y diámetro parecidos al de esta red así como el camino mínimo pero es cierto que el coeficiente de agrupamiento queda lejos del valor deseado y no hay ninguna forma de acercarlo sin alterar el resto de



propiedades.

También se puede observar que el coeficiente de agrupamiento es igual que la densidad. Esto es una característica de los grafos ER que difiere directamente con las redes reales que tiene valores distintos para estas dos propiedades. Esto quiere decir que si la red que generemos la aproximamos por coeficiente de clustering a nuestra red real la densidad de la red obtenida y el grado medio se aleja del resultado deseado y viceversa.

Ahora vamos a comprobar cómo sería una red generada con ER cuyo valor medio de agrupamiento sea aproximado al de nuestra red de estudio pokemon. Para ello tomaremos

$p=0.78$  y  $0.76$ .

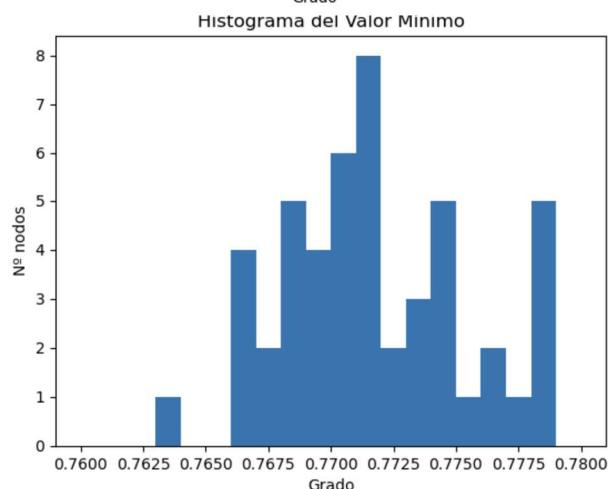
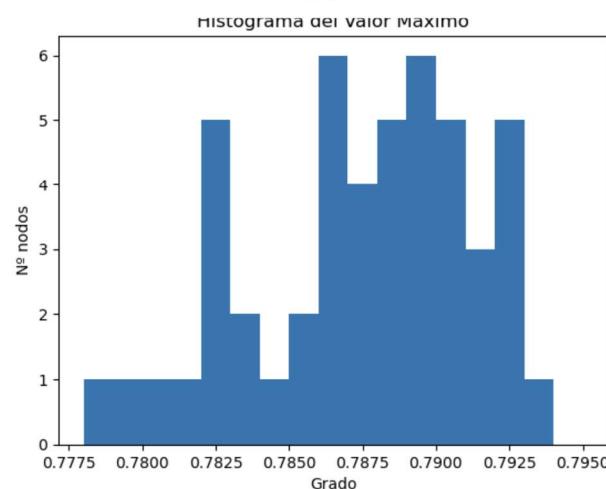
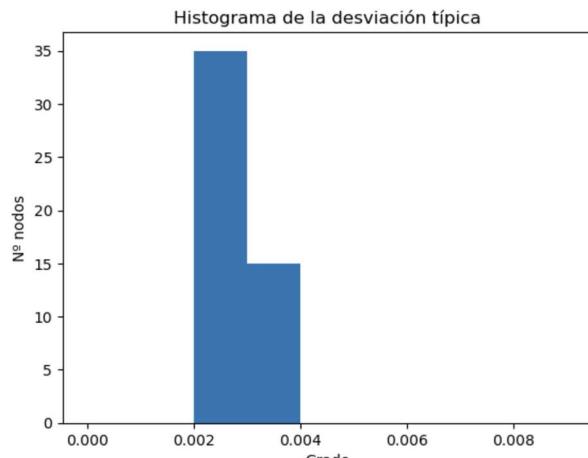
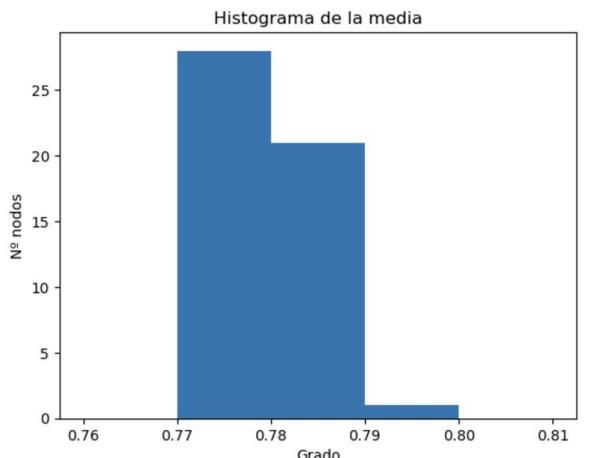
RED 1

Número de nodos: 168.0  
 Número de aristas: 10935.32  
 Camino mínimo medio: 1.2  
 Grado medio: 130.18  
 Radio medio de la red 2.0  
 Diámetro medio de la red 2.0  
 Número de componentes conexas: 1  
 Media del coeficiente de clustering 0.77956  
 Coeficiente de clustering global 0.77952  
 Densidad 0.77954

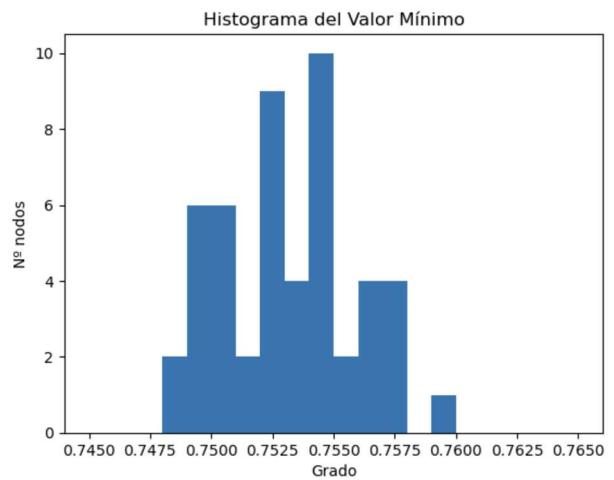
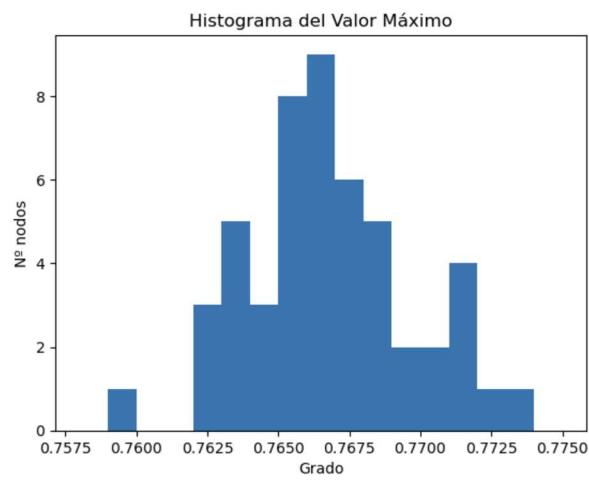
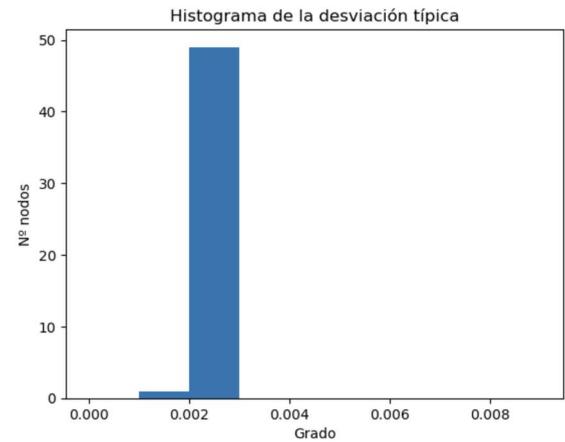
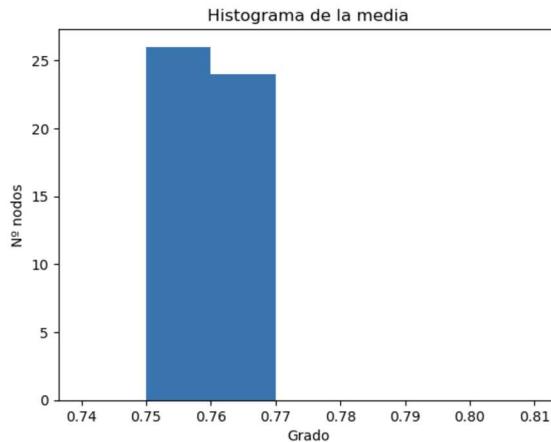
RED 1

Número de nodos: 214.0  
 Número de aristas: 17320.4  
 Camino mínimo medio: 1.2  
 Grado medio: 161.87  
 Radio medio de la red 2.0  
 Diámetro medio de la red 2.0  
 Número de componentes conexas: 1  
 Media del coeficiente de clustering 0.75992  
 Coeficiente de clustering global 0.7599  
 Densidad 0.75997

## Segunda Temporada



## Quinta Temporada

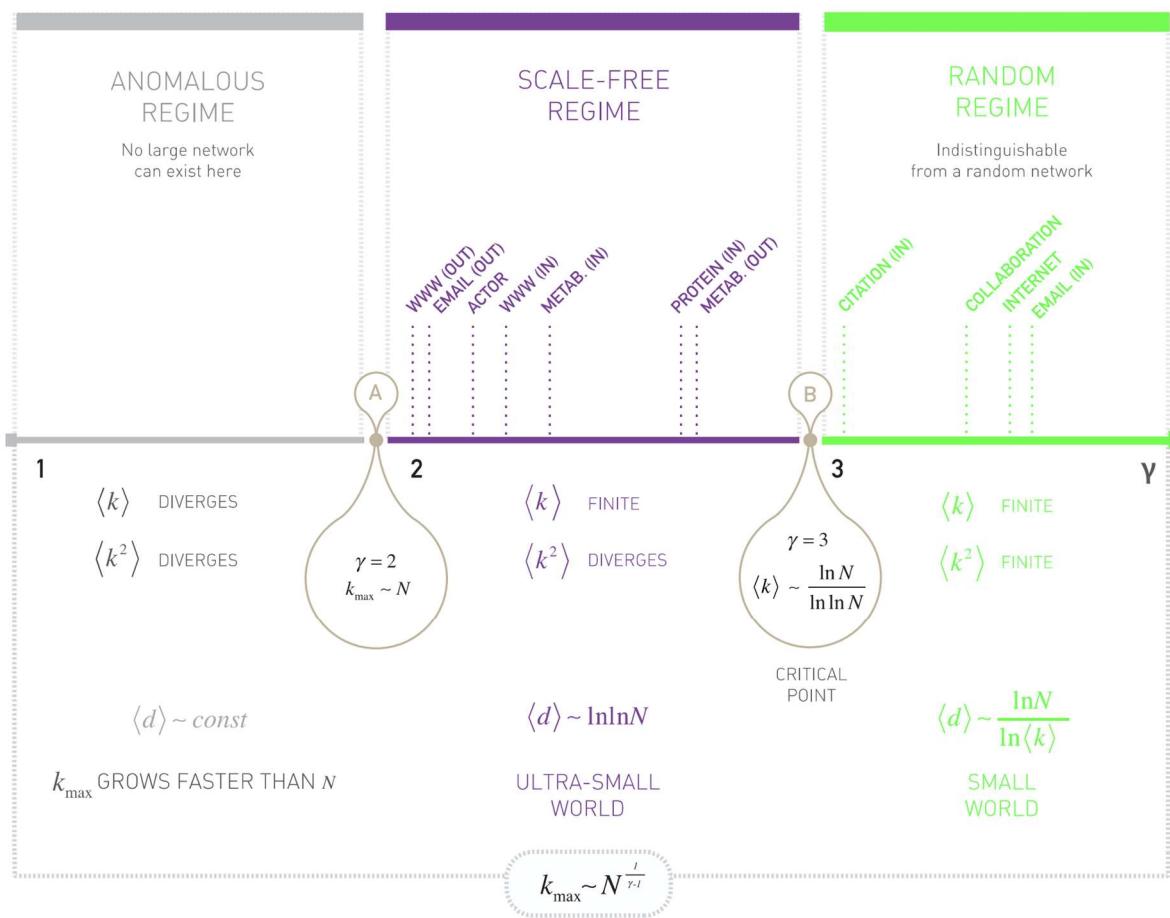


Como esperábamos obtenemos un clustering muy próximo al de nuestra red real a costa de alejar el valor de las demás características. En general podemos concluir que el modelo de ER ofrece un modelaje pobre ya que las redes reales como la nuestra pese a tener una densidad baja tienen un coeficiente de agrupamiento alto.

## Comparación con el Modelo Barabasi-Albert

En este caso ajustamos el valor de  $m$  ejecutando varias veces hasta obtener unos resultados cercanos a los de la red estudiada.

En el modelo de BA podemos calcular el camino medio teóricamente ya que este modelo se rige por la ley de mundo pequeño y su comportamiento.



Para este modelo se genera una red scale free con:  $\gamma = 3$

Por lo que está en un punto crítico entre mundo pequeño y ultra-pequeño, en este caso con valor de  $\langle k \rangle \approx 3.136$ .

Por lo general el modelo de BA tiene comportamiento de red de mundo pequeño por lo que calcularemos el camino mínimo medio con la fórmula de mundo pequeño:

Segunda Temporada

Quinta Temporada

$$\langle d \rangle = 4.483$$

$$\langle d \rangle = 4.620$$

El valor obtenido es bastante dispar al 1,63 y 1,76 de camino mínimo medio de nuestras redes. Vamos a probar con la fórmula de mundo ultra-pequeño:

Segunda Temporada

$$\langle d \rangle = 1.634$$

Quinta Temporada

$$\langle d \rangle = 1.680$$

El resultado obtenido es prácticamente idéntico a lo esperado en la segunda temporada y bastante aproximado en la quinta por lo que podemos suponer que nuestra red tiene el comportamiento de mundo ultra-pequeño.

**Segunda Temporada**

RED 1

---

Número de nodos: 168.0  
Número de aristas: 5207.0  
Camino mínimo medio: 1.6  
Grado medio: 61.99  
Radio medio de la red 2.0  
Diámetro medio de la red 2.0  
Número de componentes conexas: 1.0  
Media del coeficiente de clustering 0.47812  
Coeficiente de clustering global 0.45421

---

**Quinta Temporada**

RED 1

---

Número de nodos: 214.0  
Número de aristas: 5520.0  
Camino mínimo medio: 1.8  
Grado medio: 51.59  
Radio medio de la red 2.0  
Diámetro medio de la red 2.9  
Número de componentes conexas: 1.0  
Media del coeficiente de clustering 0.33889  
Coeficiente de clustering global 0.32968

---

Con este modelo obtenemos buenos resultados para todas las variables salvo para el coeficiente de clustering al igual que con ER pero con un resultado un poco más próximo y en este modelo sin la opción de poder aumentarlo tan considerablemente.

### Comparación con el Modelo Watts Strogatz

Este modelo intenta satisfacer las propiedades de mundos pequeños que como ya hemos visto en el modelo anterior nuestras redes se aproximan al comportamiento de estos. Puede tener un coeficiente de agrupamiento más elevado que otros modelos, justo lo que buscamos en nuestro caso. Cuanto más se aproxime el valor de p a 0 mayor será el valor del clustering pero si superamos una región crítica pierde las propiedades de mundo pequeño y se descompensarán el resto de características de las redes. Modelamos las redes para ambas temporadas buscando el mínimo valor de p para el que el coeficiente de clustering se aproxime al deseado sin alterar notablemente las demás métricas.

## Segunda Temporada

RED 1

---

Número de nodos: 168.0  
Número de aristas: 5208.0  
Camino mínimo medio: 1.6  
Grado medio: 62.0  
Radio medio de la red 2.0  
Diámetro medio de la red 3.0  
Número de componentes conexas: 1  
Media del coeficiente de clustering 0.65863  
Coeficiente de clustering global 0.6579

---

## Quinta Temporada

RED 1

---

Número de nodos: 214.0  
Número de aristas: 5564.0  
Camino mínimo medio: 1.9  
Grado medio: 52.0  
Radio medio de la red 2.0  
Diámetro medio de la red 3.0  
Número de componentes conexas: 1  
Media del coeficiente de clustering 0.64496  
Coeficiente de clustering global 0.64401

Con este modelo obtenemos unos resultados considerablemente precisos en comparación con los otros modelos. Para nuestra red real y su comportamiento resulta ser un buen modelo con el que generar redes aleatorias similares a las redes en estudio.

En general los modelos más precisos para modelar las redes que definen las temporadas de pokémon son los BA y WS ya que tratan de implementar las propiedades de mundo pequeño consiguiendo un valor del coeficiente de agrupamiento considerablemente alto como pasa con las redes reales.

## Conclusiones:

A lo largo de la memoria creo que hemos ido sacando a la luz e interpretando el grafo de forma que el proyecto se ha ido explicando así mismo a través del análisis de las diferentes métricas, por lo que ahora vamos a hacer un pequeño resumen de las conclusiones que más destacan para tener un pequeño guión de nuestro ensayo.

Finalmente tras un arduo análisis llegamos al final de la memoria donde os vamos a relatar las conclusiones a las que hemos llegado; nuestro trabajo busca hacer una interpretación sobre los grafos formados por los pokemons y la coincidencia entre ellos en diferentes episodios de estas dos temporadas, de esta forma conseguimos dos

grafos bastante parecidos después de conseguir los datos que necesitábamos a partir de hacer “web scrapping” en una wiki de pokemon y luego desarrollar un par de programas en C++ para la conversión de estos datos en archivos .csv que Gephi/Networkx pudiesen leer. Si tiene tiempo y le apetece le recomendamos que eche un ojo al edgelist converter ya que creemos que ha sido ingeniosa y sobre todo la manera que a partir de estructuras de datos hemos conseguido convertir la base de datos en lo que queríamos y en todas las utilidades e información que este programa nos da.

En estos dos grafos los nodos que más destacan por diferencia son los que en alguna ocasión he llamado como “populares”, estos son los que aparecen en todos los capítulos de la temporada como son “Pikachu”, “Meowth”, “Togepi” y “Pikachu”, “Meowth” y “Wobuffet” respectivamente ya que estos se relacionan con todos los demás pokemons y lo mágico de estos es que el peso de sus 167 aristas nos indican el número de episodios en los que ha aparecido el pokemon con quien están unidos. Esto es bastante útil para hacernos una idea de cómo de importante es un pokémon. En la quinta temporada en particular es curioso observar como “Wobuffet” tiene junto a “Pikachu” y “Meowth” un grado de 213, lo que significa que coincide en algún episodio con todo el resto de pokemons, esto es interesante ya que a diferencia de sus compañeros este no aparece en todos los episodios de la temporada pero aún así se hace muy importante. Los pokemons “populares” como es obvio debido a sus características son indiscutiblemente los nodos más importantes en la gran mayoría de métricas. Luego por el otro lado tenemos a “Magneton” y a “Xatu” que son los pokemons menos importantes de las temporadas respectivamente, estos dos pokemons aparecen en sendos episodios donde coinciden con 9 y con 6 pokemons respectivamente, esto les convierte en los pokemons más “asociales” de las temporadas.

Es curioso darnos cuenta de que los guionistas o creadores de la serie tienen una preferencia sobre los pokemons de tipo agua que son los más numerosos por su primer tipo en ambas temporadas. ¿Por qué habrán elegido este tipo como el más común los creadores de la serie?

El grado del gafo de media es muy elevado en ambas redes lo que nos dice que los pokemons coinciden con muchos pokemons en los capítulos en general, hay un elevado número de aparición de pokemons por episodios.

Para la temporada 2 tenemos:

NUM DE EPISODIOS: 63

NUM APARICIÓN POKEMONS: 873

Media aparición de pokemons por capítulos: 13.8571

Y para la temporada 5:

.NUM DE EPISODIOS: 65

NUM APARICIÓN POKEMONS: 948

Media aparición de pokemons por capítulos: 14.5846

Como vemos es ligeramente superior en la quinta temporada, esto también puede deberse a que hay más pokemons esta temporada. (El dato del número de apariciones de los pokemons lo hemos logrado haciendo un contador mientras recorremos con iteradores los pokemons que habían aparecido en cada episodio.

También mencionar que gracias a los pokemons “populares” el diámetro de ambas redes es 2, lo que hace el camino mínimo entre dos nodos o el árbol de recubrimiento mínimo enormemente pequeño.

Nuestro estudio ya que es sobre una serie de ficción realmente no tiene demasiadas aplicaciones en el mundo real, pero, es curioso darse cuenta todo lo que podemos llegar a hipotetizar o a comprender de cómo funciona la serie tras bambalinas a través del estudio e interpretación de las redes que conforman los pokemons que salen en estas. Creo que nuestro estudio aún careciendo de relevancia para resolver un problema del mundo real puede ser muy enriquecedor o de total interés para un fanático de la serie que quiera adentrarse un poco más en la serie y descubrir matices en los que nunca se había parado a pensar ya que las métricas y herramientas usadas nos ofrecen una visión sobre este tema que nunca antes creo que nadie si quiera se haya planteado. Todo esto gracias a la gran ayuda de la Teoría de grafos.

## Bibliografía:

- <https://www.geeksforgeeks.org/graph-measurements-length-distance-diameter-eccentricity-radius-center/>
- <http://gephi.wikidot.com/capitulo02b>
- [https://en.wikipedia.org/wiki/Betweenness\\_centrality](https://en.wikipedia.org/wiki/Betweenness_centrality)
- <https://www.wikidex.net/wiki/WikiDex>

- [https://www.wikidex.net/wiki/Lista\\_de\\_Pok%C3%A9mon\\_por\\_aparici%C3%B3n\\_en\\_el\\_anime](https://www.wikidex.net/wiki/Lista_de_Pok%C3%A9mon_por_aparici%C3%B3n_en_el_anime)
- [https://es.wikipedia.org/wiki/Grafo\\_conexo](https://es.wikipedia.org/wiki/Grafo_conexo)
- <https://cplusplus.com/reference/map/map/>
- <https://cplusplus.com/reference/iterator/>
- [https://hmong.es/wiki/Barab%C3%A1si%E2%80%93Albert\\_model](https://hmong.es/wiki/Barab%C3%A1si%E2%80%93Albert_model)
- NetworkX Documentation
- Numpy Documentation
- Matplotlib Documentation
- Teoría del campus virtual