

The Real Verde

Integrantes:

- Felipe Aguilera
- Matías Cárcamo
- René Fuentes
- Felipe Zabala

Requerimiento del Grupo

- Comparar la efectividad / dificultad del material. Por ejemplo, visualizar la tasa de respuestas correctas de cada problema, o el tiempo medio en resolverlo para ver que problemas son más difíciles.

Historias de Usuario

- ▶ Yo como usuario necesito ver el porcentaje de respuestas correctas de cierta actividad para ver el nivel de los alumnos en sus conocimientos.
- ▶ Yo como usuario me gustaría saber que actividades son las menos frecuentadas para así tratar de mejorarlas y que sean de mayor interés.
- ▶ Yo como usuario me gustaría saber cuál es el tiempo mínimo y máximo en resolver una actividad.

Representación de Datos JSON

```

1 Topic[
2 {"actividades": "ANIMATED_EXAMPLE", "temas": "classes_objects", "intentos_correctos": 0, "intentos_incorrectos": 0, "intentos":
3 {"actividades": "PARSONS", "temas": "classes_objects", "intentos_correctos": 91, "intentos_incorrectos": 171, "intentos": 262, "
4 {"actividades": "QUIZPET", "temas": "classes_objects", "intentos_correctos": 259, "intentos_incorrectos": 223, "intentos": 482,
5 {"actividades": "WEBEX", "temas": "classes_objects", "intentos_correctos": 0, "intentos_incorrectos": 0, "intentos": 652, "tiemp
6 {"actividades": "ANIMATED_EXAMPLE", "temas": "Comparison", "intentos_correctos": 0, "intentos_incorrectos": 0, "intentos": 1356,
7 {"actividades": "PARSONS", "temas": "Comparison", "intentos_correctos": 573, "intentos_incorrectos": 231, "intentos": 804, "tiem
8 {"actividades": "QUIZPET", "temas": "Comparison", "intentos_correctos": 306, "intentos_incorrectos": 264, "intentos": 570, "tiem
9 {"actividades": "WEBEX", "temas": "Comparison", "intentos_correctos": 0, "intentos_incorrectos": 0, "intentos": 891, "tiempo_pro
10 {"actividades": "ANIMATED_EXAMPLE", "temas": "dictionary", "intentos_correctos": 0, "intentos_incorrectos": 0, "intentos": 1781,
11 {"actividades": "PARSONS", "temas": "dictionary", "intentos_correctos": 228, "intentos_incorrectos": 88, "intentos": 316, "tiemp
12 {"actividades": "QUIZPET", "temas": "dictionary", "intentos_correctos": 152, "intentos_incorrectos": 66, "intentos": 218, "tiemp
13 {"actividades": "WEBEX", "temas": "dictionary", "intentos_correctos": 0, "intentos_incorrectos": 0, "intentos": 328, "tiempo_pro
14 {"actividades": "ANIMATED_EXAMPLE", "temas": "exceptions", "intentos_correctos": 0, "intentos_incorrectos": 0, "intentos": 330,
15 {"actividades": "PARSONS", "temas": "exceptions", "intentos_correctos": 61, "intentos_incorrectos": 1140, "intentos": 1201, "tie
16 {"actividades": "QUIZPET", "temas": "exceptions", "intentos_correctos": 127, "intentos_incorrectos": 82, "intentos": 209, "tiemp
17 {"actividades": "WEBEX", "temas": "exceptions", "intentos_correctos": 0, "intentos_incorrectos": 0, "intentos": 446, "tiempo_pro

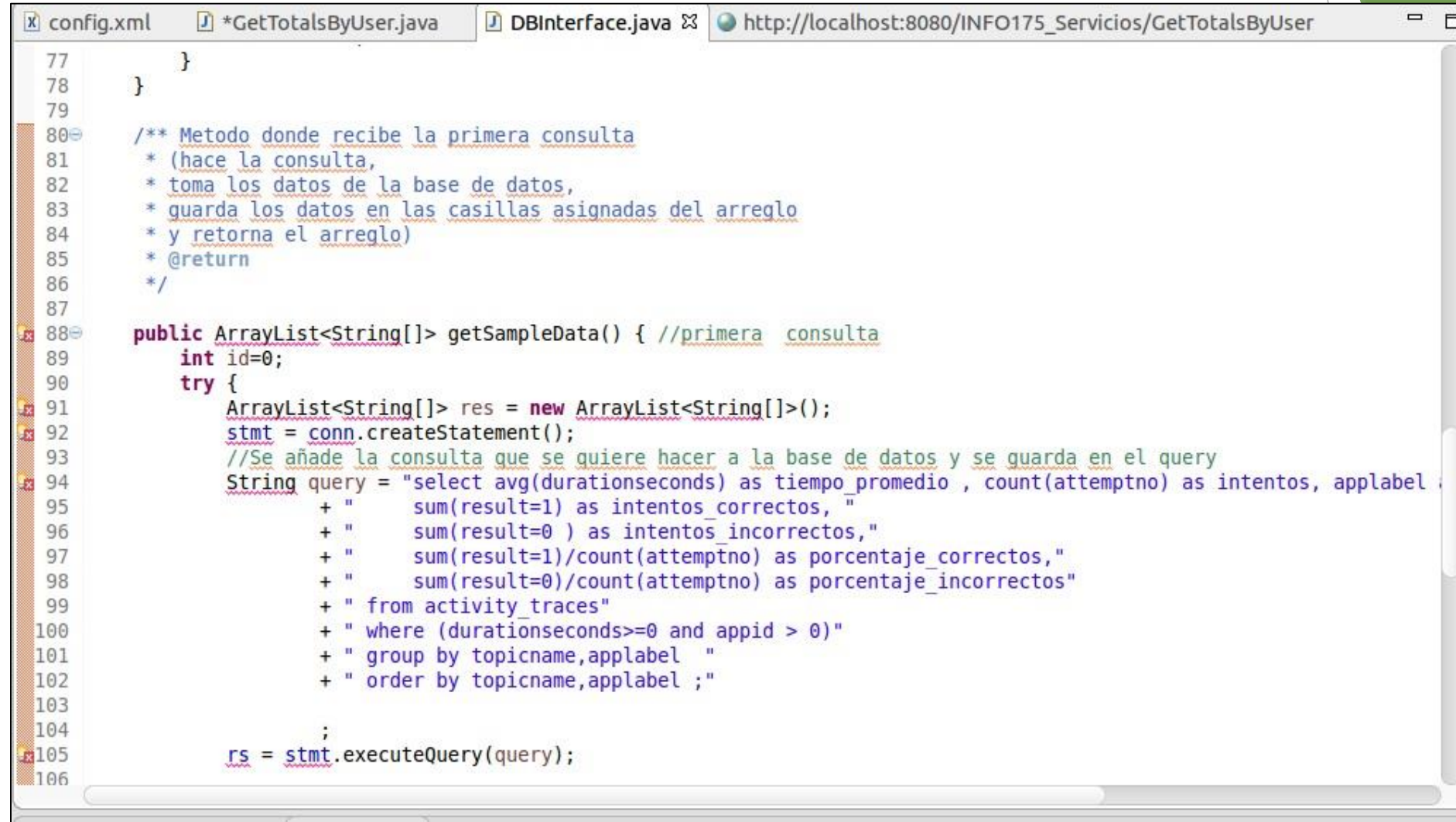
```

```

"intentos": 1756, "tiempo_promedio": 6.0979, "porcentaje_correctos": 0.0000, "porcentaje_incorrectos": 0.0000},
ntos": 262, "tiempo_promedio": 36.9809, "porcentaje_correctos": 0.3473, "porcentaje_incorrectos": 0.6527},
entos": 482, "tiempo_promedio": 125.3983, "porcentaje_correctos": 0.5373, "porcentaje_incorrectos": 0.4627},
: 652, "tiempo_promedio": 10.3512, "porcentaje_correctos": 0.0000, "porcentaje_incorrectos": 0.0000},
entos": 1356, "tiempo_promedio": 5.2832, "porcentaje_correctos": 0.0000, "porcentaje_incorrectos": 0.0000},
": 804, "tiempo_promedio": 24.9030, "porcentaje_correctos": 0.7127, "porcentaje_incorrectos": 0.2873},
": 570, "tiempo_promedio": 39.0351, "porcentaje_correctos": 0.5368, "porcentaje_incorrectos": 0.4632},
, "tiempo_promedio": 2.5342, "porcentaje_correctos": 0.0000, "porcentaje_incorrectos": 0.0000},
entos": 1781, "tiempo_promedio": 5.0180, "porcentaje_correctos": 0.0000, "porcentaje_incorrectos": 0.0000},
: 316, "tiempo_promedio": 30.6930, "porcentaje_correctos": 0.7215, "porcentaje_incorrectos": 0.2785},
: 218, "tiempo_promedio": 49.9817, "porcentaje_correctos": 0.6972, "porcentaje_incorrectos": 0.3028},
, "tiempo_promedio": 3.6250, "porcentaje_correctos": 0.0000, "porcentaje_incorrectos": 0.0000},
entos": 330, "tiempo_promedio": 4.5788, "porcentaje_correctos": 0.0000, "porcentaje_incorrectos": 0.0000},
": 1201, "tiempo_promedio": 35.2831, "porcentaje_correctos": 0.0508, "porcentaje_incorrectos": 0.9492},
: 209, "tiempo_promedio": 53.7177, "porcentaje_correctos": 0.6077, "porcentaje_incorrectos": 0.3923},
, "tiempo_promedio": 6.6570, "porcentaje_correctos": 0.0000, "porcentaje_incorrectos": 0.0000},
intentos": 1557, "tiempo_promedio": 4.2877, "porcentaje_correctos": 0.0000, "porcentaje_incorrectos": 0.0000},

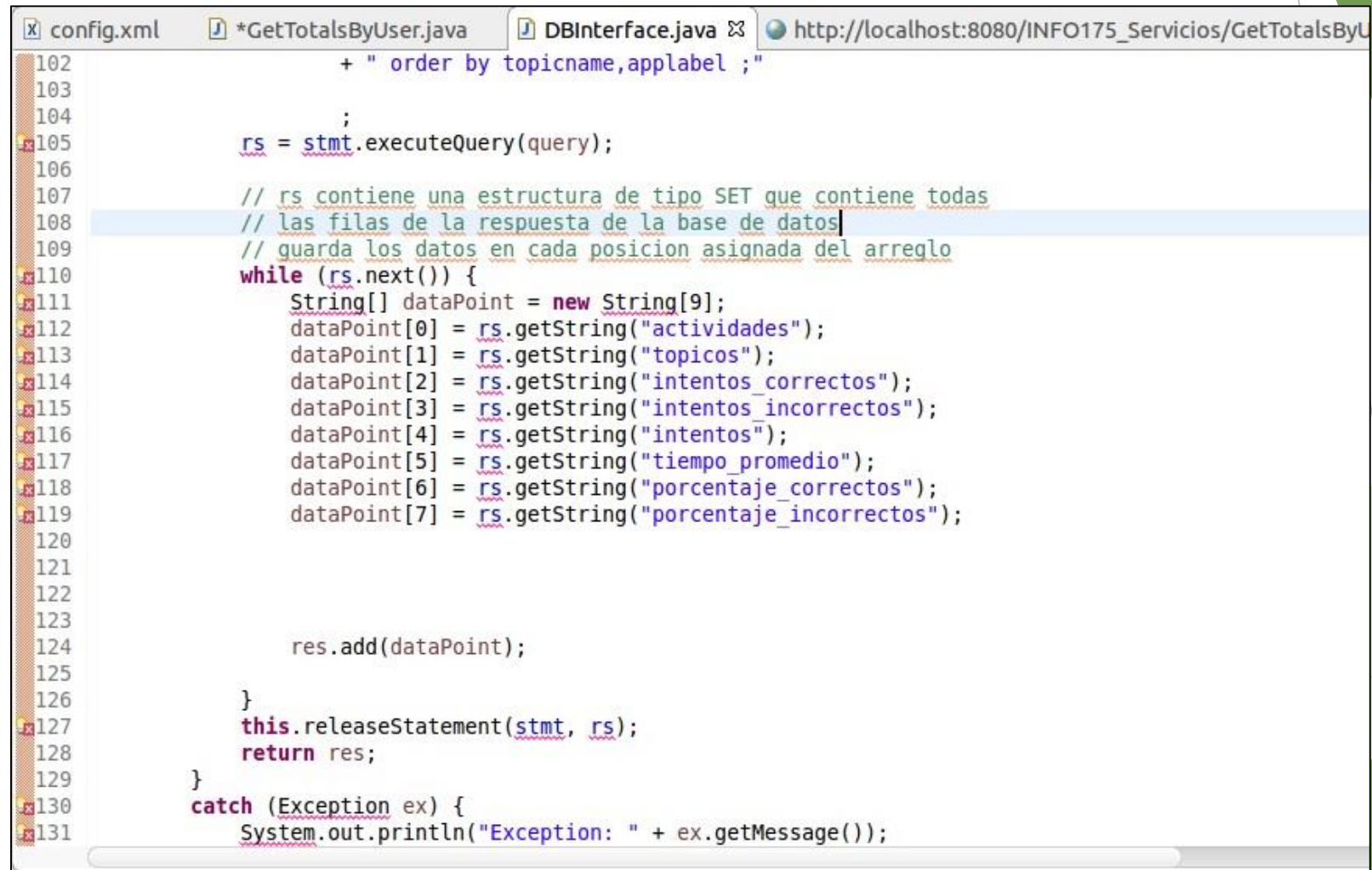
```


Código JSON



```
77     }
78 }
79
80 /** Metodo donde recibe la primera consulta
81  * (hace la consulta,
82  * toma los datos de la base de datos,
83  * guarda los datos en las casillas asignadas del arreglo
84  * y retorna el arreglo)
85  * @return
86  */
87
88 public ArrayList<String[]> getSampleData() { //primera consulta
89     int id=0;
90     try {
91         ArrayList<String[]> res = new ArrayList<String[]>();
92         stmt = conn.createStatement();
93         //Se añade la consulta que se quiere hacer a la base de datos y se guarda en el query
94         String query = "select avg(durationseconds) as tiempo_promedio , count(attemptno) as intentos, applabel
95             + "      sum(result=1) as intentos_correctos, "
96             + "      sum(result=0 ) as intentos_incorrectos,"
97             + "      sum(result=1)/count(attemptno) as porcentaje_correctos,"
98             + "      sum(result=0)/count(attemptno) as porcentaje_incorrectos"
99             + " from activity_traces"
100             + " where (durationseconds>=0 and appid > 0)"
101             + " group by topicname,applabel "
102             + " order by topicname,applabel ;"
103
104         ;
105         rs = stmt.executeQuery(query);
106     }
```

Código JSON

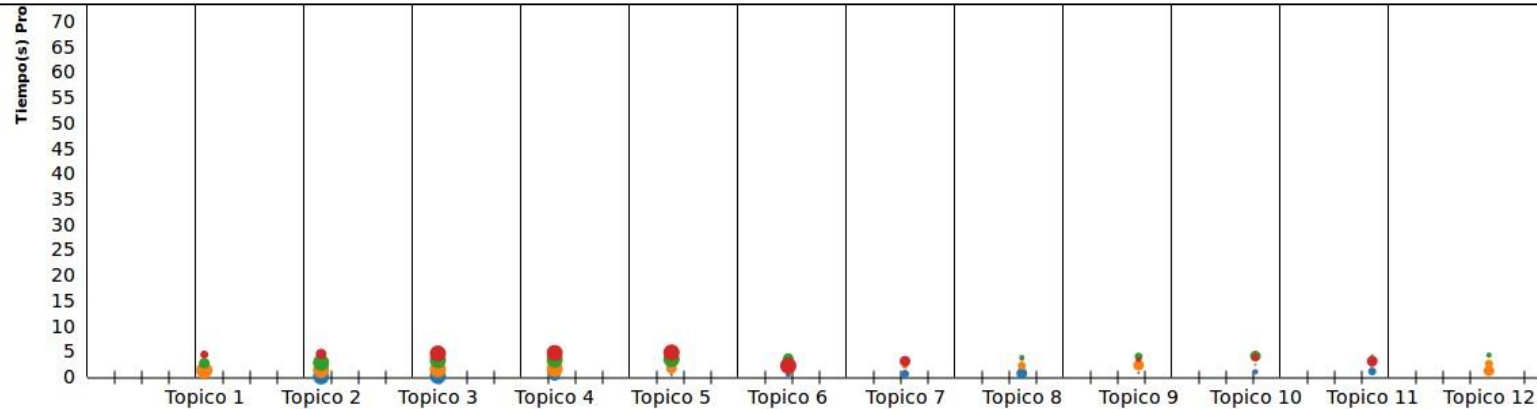


```
102         + " order by topicname,applabel ;"
103
104         ;
105         rs = stmt.executeQuery(query);
106
107         // rs contiene una estructura de tipo SET que contiene todas
108         // las filas de la respuesta de la base de datos
109         // guarda los datos en cada posicion asignada del arreglo
110         while (rs.next()) {
111             String[] dataPoint = new String[9];
112             dataPoint[0] = rs.getString("actividades");
113             dataPoint[1] = rs.getString("temas");
114             dataPoint[2] = rs.getString("intentos_correctos");
115             dataPoint[3] = rs.getString("intentos_incorrectos");
116             dataPoint[4] = rs.getString("intentos");
117             dataPoint[5] = rs.getString("tiempo_promedio");
118             dataPoint[6] = rs.getString("porcentaje_correctos");
119             dataPoint[7] = rs.getString("porcentaje_incorrectos");
120
121
122
123
124             res.add(dataPoint);
125
126         }
127         this.releaseStatement(stmt, rs);
128         return res;
129     }
130     catch (Exception ex) {
131         System.out.println("Exception: " + ex.getMessage());
```

Código JSON

```
config.xml  *GetTotalsByUser.java  DBInterface.java  http://localhost:8080/INFO175_Servicios/GetTotalsByUser
50      out.println(outAsJSON4(data1));
57      |
58      }
59
60      protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
61          doGet(request, response);
62      }
63      private String outAsJSON4(ArrayList<String[]> data1) {
64          String outString="Topic[";
65          for(String[] row : data1) {
66              outString += "\n "
67                  + "{ \"actividades\": \"" + row[0] //datos consulta
68                  + "\", \"temas\": \"" + row[1]
69                  + "\", \"intentos_correctos\": " + row[2]
70                  + ", \"intentos_incorrectos\": " + row[3]
71                  + ", \"intentos\": " + row[4]
72                  + ", \"tiempo_promedio\": " + row[5]
73                  + ", \"porcentaje_correctos\": " + row[6]
74                  + ", \"porcentaje_incorrectos\": " + row[7]
75
76                  + "}, ";
77          }
78
79          outString = outString.substring(0, outString.length() - 1)+"\n ]";
80
81          return outString;
82      }
83
84      }
85      --
```

Visualización de Datos



Actividades

- ☒ Questions
- ☒ Parsons
- ☒ Webex
- ☒ Animated

Leyenda

Actividades con dificultad

MD D I F MF

Question (Q)

MD D I F MF

Parsons (P)

Actividades sin dificultad

Webex (W)

Animated Example(AE)

-Tamaño: Frecuencia en la Actividad

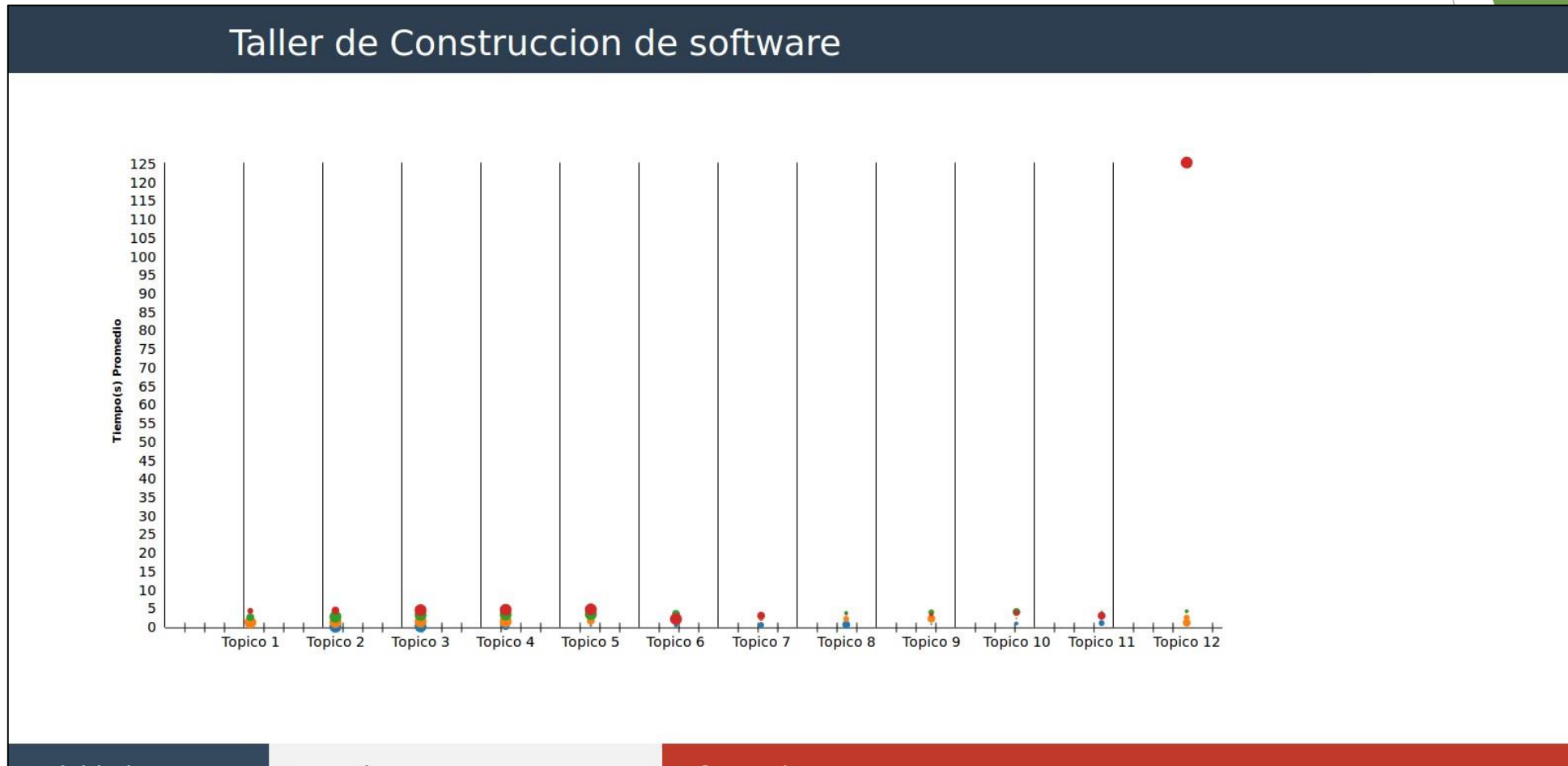
-Eje ordenada: Tiempo Promedio en la Actividad

-Eje Abscisa: Topicos y Actividades

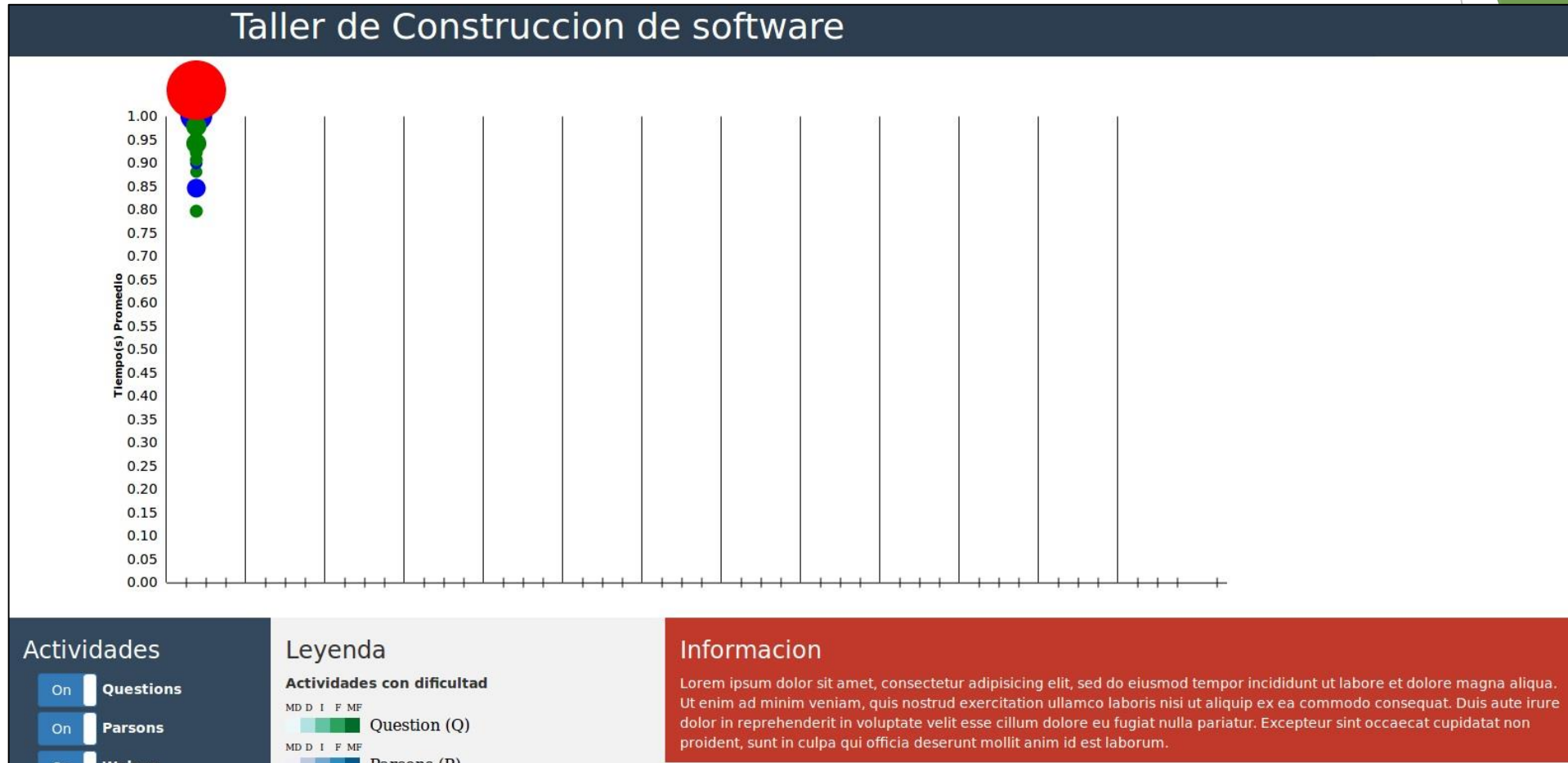
Informacion

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Visualización de Datos



Visualización de Datos



Código Visualización

```
34 var outerWidth = 1230; // ancho exterior del grafico
35 var outerHeight = 560; // altura exterior del grafico
36 var margin = { left: 80, top: 30, right: 30, bottom: 30 }; // margenes desde los bordes de la ventana del grafico
37 var barPadding = 2; // relleno barra
38
39 var xColumn = "topic"; // nombre de la columna con los topicos en el archivo csv
40 var yColumn = "time"; // nombre de la columna cn el tiempo en el archivo csv
41 var colorColumn = "color"; // nombre de la columna en el archivo csv, dificultad, mas degradado mas facil
42 var rColumn = "radio"; // nombre de la columna en el archivo csv, a mayor radio mas frecuentada la actividad
43 var rMin=1; // radio minimo
44 var rMax=6; // radio maximo
45
46 var xAxisLabelText = "Topicos"; // nombre abscisa grafico
47 var xAxisLabelOffset = 48; // no se ocupa
48
49 var yAxisLabelText = "Tiempo(s) Promedio"; // nombre ordenada grafico
50 var yAxisLabelOffset = 45; // ubicacion del texto en el eje X
51
52 var innerWidth = outerWidth - margin.left - margin.right; // ancho interior
53 var innerHeight = outerHeight - margin.top - margin.bottom; // altura interior
54
55
56 /// se agrega al cuerpo el ancho y el alto de la ventana de visualizacion
57 var svg = d3.select("body").append("svg")
58     .attr("width", outerWidth)
59     .attr("height", outerHeight);
60
```

Código Visualización

```
index.html {} data.json grafico.js estilos.css
60
61 // ubicar grafico en ventana
62 var g = svg.append("g")
63   .attr("transform", "translate(" + margin.left * 2 + "," + margin.top + ")");
64
65 // ubica el eje de las abscisa (X)
66 var xAxisG = g.append("g")
67   .attr("class", "x axis")
68   .attr("transform", "translate(0," + innerHeight + ")");
69
70
71 var yAxisG = g.append("g")
72   .attr("class", "y axis");
73   // .attr("transform", "translate(15,0)");
74
75 /// crear divisores en el grafico
76 var yAxisG2 = g.append("g").attr("class", "y2 axis").attr("transform", "translate(80,0)");
77 var yAxisG3 = g.append("g").attr("class", "y3 axis").attr("transform", "translate(160,0)");
78 var yAxisG4 = g.append("g").attr("class", "y4 axis").attr("transform", "translate(240,0)");
79 var yAxisG5 = g.append("g").attr("class", "y5 axis").attr("transform", "translate(320,0)");
80 var yAxisG6 = g.append("g").attr("class", "y6 axis").attr("transform", "translate(400,0)");
81 var yAxisG7 = g.append("g").attr("class", "y7 axis").attr("transform", "translate(480,0)");
82 var yAxisG8 = g.append("g").attr("class", "y8 axis").attr("transform", "translate(560,0)");
83 var yAxisG9 = g.append("g").attr("class", "y9 axis").attr("transform", "translate(640,0)");
84 var yAxisG10 = g.append("g").attr("class", "y10 axis").attr("transform", "translate(720,0)");
85 var yAxisG11 = g.append("g").attr("class", "y11 axis").attr("transform", "translate(800,0)");
86 var yAxisG12 = g.append("g").attr("class", "y12 axis").attr("transform", "translate(880,0)");
87 var yAxisG13 = g.append("g").attr("class", "y13 axis").attr("transform", "translate(960,0)");
88 var yAxisG14 = g.append("g").attr("class", "y14 axis").attr("transform", "translate(740,0)");
89
90
91 // escalan los valores en el grafico
```


Código Visualización

```
index.html {} data.json grafico.js estilos.css
90
91 // escalan los valores en el grafico
92 var xScale = d3.scale.ordinal().rangeBands([0, innerWidth], barPadding);
93 var yScale = d3.scale.linear().range([innerHeight,30]);
94 //var rScale = d3.scale.linear().range([rMin, rMax]);
95 var colorScale = d3.scale.category10();
96
97
98 /*
99 var x = d3.scale.linear().domain([0, 25]).range([0, width]);
100 var data = [1, 2, 3, 5, 8, 13, 21];
101 var xAxis = d3.svg.axis().scale(x).orient("top").tickValues(data).innerTickSize([250]).outerTickSize([250]);
102 */
103
104 // se agregan lineas en el eje X, desde el eje a la orientacion
105 var xAxis = d3.svg.axis().scale(xScale).orient("bottom").outerTickSize(1); // Turn off the marks at the end of the axis.
106
107 // crea las marcas en el eje Y con respecto a los datos, en este caso al tiempo
108 // si el tiempo es 150 seg dividira 125/20 = 6.25 aproxima a 5, cada 5 hara una marca
109 //
110 var yAxis = d3.svg.axis().scale(yScale).orient("left")
111     .ticks(20) // Use approximately 20 ticks marks.
112     //.tickFormat(d3.format("d")) // Use intelligent abbreviations, e.g. 5M for 5 Million
113     .outerTickSize(1); // Turn off the marks at the end of the axis.
114
115 //orientacion de las lineas
116 var yAxis2 = d3.svg.axis().scale(yScale).orient("left").ticks(0).outerTickSize(1);
117 var yAxis3 = d3.svg.axis().scale(yScale).orient("left").ticks(0).outerTickSize(1);
118 var yAxis4 = d3.svg.axis().scale(yScale).orient("left").ticks(0).outerTickSize(1);
119 var yAxis5 = d3.svg.axis().scale(yScale).orient("left").ticks(0).outerTickSize(1);
120 var yAxis6 = d3.svg.axis().scale(yScale).orient("left").ticks(0).outerTickSize(1);
121 var yAxis7 = d3.svg.axis().scale(yScale).orient("left").ticks(0).outerTickSize(1);
122 var yAxis8 = d3.svg.axis().scale(yScale).orient("left").ticks(0).outerTickSize(1);
```

Código Visualización

```
index.html {} data.json grafico.js estilos.css
114
115 ///orientacion de las lineas
116 var yAxis2 = d3.svg.axis().scale(yScale).orient("left").ticks(0).outerTickSize(1);
117 var yAxis3 = d3.svg.axis().scale(yScale).orient("left").ticks(0).outerTickSize(1);
118 var yAxis4 = d3.svg.axis().scale(yScale).orient("left").ticks(0).outerTickSize(1);
119 var yAxis5 = d3.svg.axis().scale(yScale).orient("left").ticks(0).outerTickSize(1);
120 var yAxis6 = d3.svg.axis().scale(yScale).orient("left").ticks(0).outerTickSize(1);
121 var yAxis7 = d3.svg.axis().scale(yScale).orient("left").ticks(0).outerTickSize(1);
122 var yAxis8 = d3.svg.axis().scale(yScale).orient("left").ticks(0).outerTickSize(1);
123 var yAxis9 = d3.svg.axis().scale(yScale).orient("left").ticks(0).outerTickSize(1);
124 var yAxis10 = d3.svg.axis().scale(yScale).orient("left").ticks(0).outerTickSize(1);
125 var yAxis11 = d3.svg.axis().scale(yScale).orient("left").ticks(0).outerTickSize(1);
126 var yAxis12 = d3.svg.axis().scale(yScale).orient("left").ticks(0).outerTickSize(1);
127 var yAxis13 = d3.svg.axis().scale(yScale).orient("left").ticks(0).outerTickSize(1);
128 var yAxis13 = d3.svg.axis().scale(yScale).orient("left").ticks(0).outerTickSize(1);
129
130
131
132 var yAxisG = g.append("g")
133   .attr("class", "y axis")
134   .attr("transform", "translate(0,0)");
135
136 // ubicacion del texto del eje Y
137 var yAxisLabel = yAxisG.append("text")
138   .style("text-anchor", "middle")
139   .attr("transform", "translate(-" + yAxisLabelOffset + ", " + (innerHeight / 2) + ") rotate(-90)")
140   .attr("class", "label")
141   .text(yAxisLabelText);
142
143 // dibuja las marcas divisoras(subzonas)
144 var cl=4;
145 for (var i = 20; i < innerWidth; i=i+20 ) {
```

Código Visualización

```
index.html {} data.json grafico.js estilos.css
125 var yAxis11 = d3.svg.axis().scale(yScale).orient("left").ticks(0).outerTickSize(1);
126 var yAxis12 = d3.svg.axis().scale(yScale).orient("left").ticks(0).outerTickSize(1);
127 var yAxis13 = d3.svg.axis().scale(yScale).orient("left").ticks(0).outerTickSize(1);
128 var yAxis13 = d3.svg.axis().scale(yScale).orient("left").ticks(0).outerTickSize(1);
129
130
131
132 var yAxisG = g.append("g")
133   .attr("class", "y axis")
134   .attr("transform", "translate(0,0)");
135
136 // ubicacion del texto del eje Y
137 var yAxisLabel = yAxisG.append("text")
138   .style("text-anchor", "middle")
139   .attr("transform", "translate(-" + yAxisLabelOffset + ", " + (innerHeight / 2) + ") rotate(-90)")
140   .attr("class", "label")
141   .text(yAxisLabelText);
142
143 // dibuja las marcas divisoras(subzonas)
144 var cl=4;
145 for (var i = 20; i < innerWidth; i=i+20 ) {
146   if(i != cl*20){
147     var linea = g.append("line").attr("x1",i).attr("y1",495).attr("x2",i).attr("y2",505).attr("stroke-width",1);
148   }
149   else{
150     cl=cl+4;
151   }
152 }
153
154
155
156
157
```


Código Visualización



```
index.html {} data.json grafico.js estilos.css
189 */
190
191 function render(data){
192
193
194 var minIntentos = d3.min(data, function (d){return d.intentos});
195 var maxIntentos = d3.max(data, function (d){return d.intentos});
196 var rScale = d3.scale.linear().domain([minIntentos, maxIntentos]).range([5,30]);
197
198 var circles = g.selectAll("circle")
199     .data(data)
200     .enter()
201     .append("circle");
202
203 var circleAttributes = circles
204     .attr("cx", 30)
205     .attr("cy", function (d) { return d.tiempo_promedio; })
206     .attr("r", function (d) { return rScale(d.intentos); })
207     .style("fill", function(d) {
208         var returnColor;
209         if(d.actividades == "ANIMATED_EXAMPLE"){
210             returnColor = "purple";
211         }else if(d.actividades == "WEBEX"){
212             returnColor = "red";
213         }else if(d.actividades == "PARSONS"){
214             returnColor = "blue";
215         }else {
216             returnColor = "green";
217         }
218         return returnColor;
219     });
220
```


Código Visualización

```
index.html {} data.json grafico.js estilos.css
205 .attr("cx", 50);
206 .attr("cy", function (d) { return d.tiempo_promedio; })
207 .attr("r", function (d) { return rScale(d.intentos); })
208 .style("fill", function(d) {
209     var returnColor;
210     if(d.actividades == "ANIMATED_EXAMPLE"){
211         returnColor = "purple";
212     }else if(d.actividades == "WEBEX"){
213         returnColor = "red";
214     }else if(d.actividades == "PARSONS"){
215         returnColor = "blue";
216     }else {
217         returnColor = "green";
218     }
219     return returnColor;
220 });
221 xAxisG.call(xAxis);
222 yAxisG.call(yAxis);
223 yAxisG2.call(yAxis2);
224 yAxisG3.call(yAxis3);
225 yAxisG4.call(yAxis4);
226 yAxisG5.call(yAxis5);
227 yAxisG6.call(yAxis6);
228 yAxisG7.call(yAxis7);
229 yAxisG8.call(yAxis8);
230 yAxisG9.call(yAxis9);
231 yAxisG10.call(yAxis10);
232 yAxisG11.call(yAxis11);
233 yAxisG12.call(yAxis12);
234 yAxisG13.call(yAxis13);
235 yAxisG14.call(yAxis14);
236 }
```