

Informe de encuestas con gráficos

Chart.js

Para la realización de este A+, utilizaremos Chart.js (<https://www.chartjs.org/>), una poderosa librería JavaScript para la visualización de datos, que nos proporciona múltiples formas gráficas a elegir (gráfico de barras, gráfico circular, gráfico de líneas...), además de diversos colores y configuraciones para las descripciones emergentes de texto.

Esta librería utiliza el elemento HTML canvas, incorporado en HTML 5, que permite la renderización interpretada dinámica de gráficos 2D y mapas de bits, así como animaciones con estos gráficos. Es por ello que su compatibilidad no es completa, funciona en los navegadores:

- Chrome 50+
- Firefox 45+
- Internet Explorer 11
- Edge 14+
- Safari 9+

En nuestro caso, simplificamos su uso utilizando solo los gráficos de barras en los que no tenemos que preocuparnos por definir los colores y cuyo uso está probado en el navegador Chrome que nos proporcionan.

Controlador

En el controlador de encuesta (SurveyController.java), debemos obtener las preguntas de una encuesta y por cada una de esta, sus respuestas y el ratio del número de veces que se ha escogido esa respuesta, con respecto al total de veces que se han escogido alguna respuesta de la pregunta.

Para ello, creamos un Map<Question, Map<Answer, Double>>, donde le pasamos las preguntas de la encuesta en concreto, por cada una sus respuestas y su ratio. Por otro lado, le pasamos model, que sería el tipo de actor logueado, puede ser Company, Moderator o Sponsor.

```
// Display
@RequestMapping(value="/display", method = RequestMethod.GET)
public ModelAndView display(@RequestParam int surveyId, @PathVariable(value="actor") final String model) {
    ModelAndView result;
    Survey survey;
    Collection<Question> questions;
    Collection<Answer> answers;
    LinkedHashMap <Answer, Double> answersRatio;
    LinkedHashMap <Question, Map<Answer, Double>> questionsAnswerRatio;

    survey = this.surveyService.findOneToEdit(surveyId);
    Assert.notNull(survey);

    questions = this.questionService.findBySurveyId(surveyId, 1,
        this.questionService.countBySurveyId(surveyId)).getContent();

    questionsAnswerRatio = new LinkedHashMap <Question, Map<Answer, Double>>();

    for(Question q : questions) {
        answersRatio = new LinkedHashMap <Answer, Double>();
        answers = this.answerService.findByQuestionId(q.getId());
        for (Answer a : answers)
            answersRatio.put(a, this.answerService.ratioAnswerPerQuestion(q.getId(), a.getId()));
        questionsAnswerRatio.put(q, answersRatio);
    }

    result = new ModelAndView("survey/display");
    result.addObject("requestURI", "survey/"+model+"/list.do");
    result.addObject("survey", survey);
    result.addObject("questions", questions);
    result.addObject("questionsAnswerRatio", questionsAnswerRatio);

    return result;
}
```

Vista

En primer lugar, debemos definir donde se encuentra el script de Chart.js, que podemos descargar desde la URL: <https://github.com/chartjs/Chart.js/releases>.

Recorremos el primer mapa y mostramos el texto de cada pregunta. Creamos el objeto HTML canvas donde se incluirá la gráfica de las respuestas de esa pregunta, nos aseguramos de ello asignando el número de la pregunta como id del objeto. Asignamos un mensaje internacionalizado a la variable “mediaRespuestas” para incluirlo luego en la gráfica.

```
<script src="scripts/Chart.js"></script>
<div>

    <jstl:forEach items="${questionsAnswerRatio}" var="mapEntry">

        <h2><jstl:out value="${mapEntry.key.text}" /></h2>

        <canvas id="<jstl:out value="${mapEntry.key.number}" />"></canvas>
        <spring:message code="survey.avg.answer" var="mediaRespuestas"/>
```

A continuación, creamos el script necesario para cada gráfica. Recorremos el segundo mapa, y asignamos como etiquetas el texto de las respuestas de cada pregunta. Ahora definimos los datos y parámetros de la gráfica:

- label: la variable antes definida como “mediaRespuestas”.
- Un conjunto de variables como fill, lineTension, backgroundColor, borderColor, borderCapStyle o borderDash (entre otras), que servirán para definir los colores usados (en nuestro caso utilizaremos un azul claro, en concreto rgba(75,192,192,1) y tonos blancos como rgba(75,192,192,1) o #fff. Por otro lado también definiremos el tamaño de los bordes, de las líneas o de los puntos, en los que estableceremos valores bajos para mejorar la apariencia.
- data: serán el ratio de cada respuesta que obtendremos al recorrer el otro mapa.

```
<canvas id="<jstl:out value="${mapEntry.key.number}" />" /></canvas>
<spring:message code="survey.avg.answer" var="mediaRespuestas"/>

<script>
var ctx = document.getElementById('<jstl:out value="${mapEntry.key.number}" />');
var media = new Chart.Bar(ctx, {
  data: {
    labels: [<jstl:forEach items="${mapEntry.value}" var="mapValue"><jstl:out value="\'" escapeXml="false"/><jstl:out value="${mapValue.key.text}"/><jstl:out value="\'" escapeXml="false"/></jstl:forEach>],
    datasets: [{
      label: '${mediaRespuestas}',
      fill: true,
      lineTension: 0.5,
      backgroundColor: "rgba(75,192,192,0.4)",
      borderColor: "rgba(75,192,192,1)",
      borderCapStyle: 'butt',
      borderDash: [],
      borderDashOffset: 0.0,
      borderJoinStyle: 'miter',
      pointBorderColor: "rgba(75,192,192,1)",
      pointBackgroundColor: "#fff",
      pointBorderWidth: 1,
      pointHoverRadius: 5,
      pointHoverBackgroundColor: "rgba(75,192,192,1)",
      pointHoverBorderColor: "rgba(220,220,220,1)",
      pointHoverBorderWidth: 1,
      pointRadius: 5,
      pointHitRadius: 10,
      data: [<jstl:forEach items="${mapEntry.value}" var="mapValue"><jstl:out value="\'" escapeXml="false"/>
<jstl:out value="${mapValue.value}"/><jstl:out value="\'" escapeXml="false"/></jstl:forEach>],
      spanGaps: true,
    }]
  }
});
</script>

</jstl:forEach>
</div>
```

Resultados

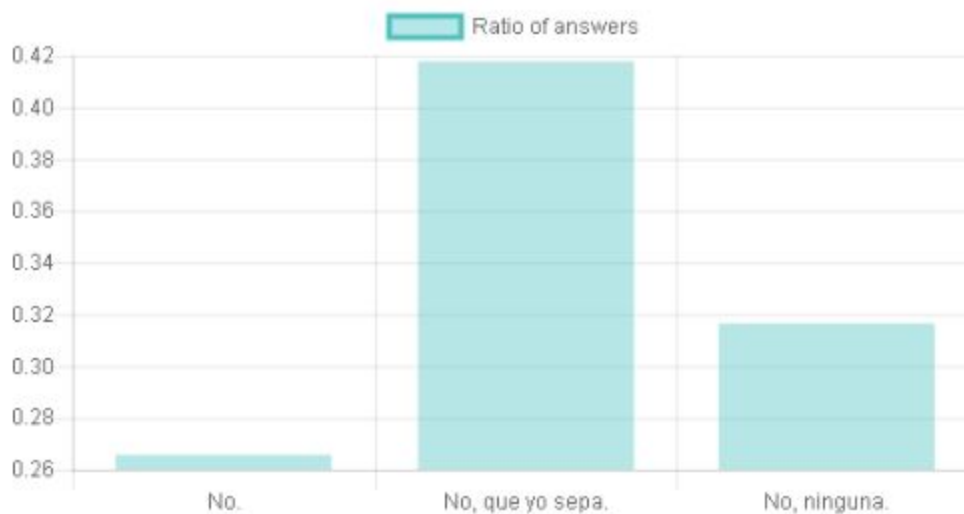
De esta forma, tras realizar insertar, crear y cambiar los archivos necesarios, obtenemos las gráficas de líneas de las que hemos hablado. Si pasamos el ratón por cada punto de la gráfica, nos aparece una descripción emergente con el nombre de la respuesta y su ratio.

Survey

¿Tienes alergia a algo?



¿Sufres alguna enfermedad cardiaca?



De esta forma, ofrecemos al creador de la encuesta, una forma mucho más sencilla y visual de conocer cuales son las respuestas que más eligen los usuarios o patrocinadores.