

Procesos Batch

- Los procesos batch o procesos por lotes son programas que suelen ser ejecutados sin ningún tipo de Intervención humana. Estos procesos son relativamente pesados ya que generalmente manipulan un volumen considerable de información y su horario de ejecución suele ocurrir en momento donde la carga o procesamiento de información es bajo, Esto con el objetivo que estos procesos no afecte el entorno transaccional

Ejemplos de procesos batch

- ▶ Procesos de generación de pre facturas
- ▶ Procesos de migración de una base de datos a otra

Que nos provee Spring Framework para trabajar con procesos batchs

Spring Boot



Spring Batch

+



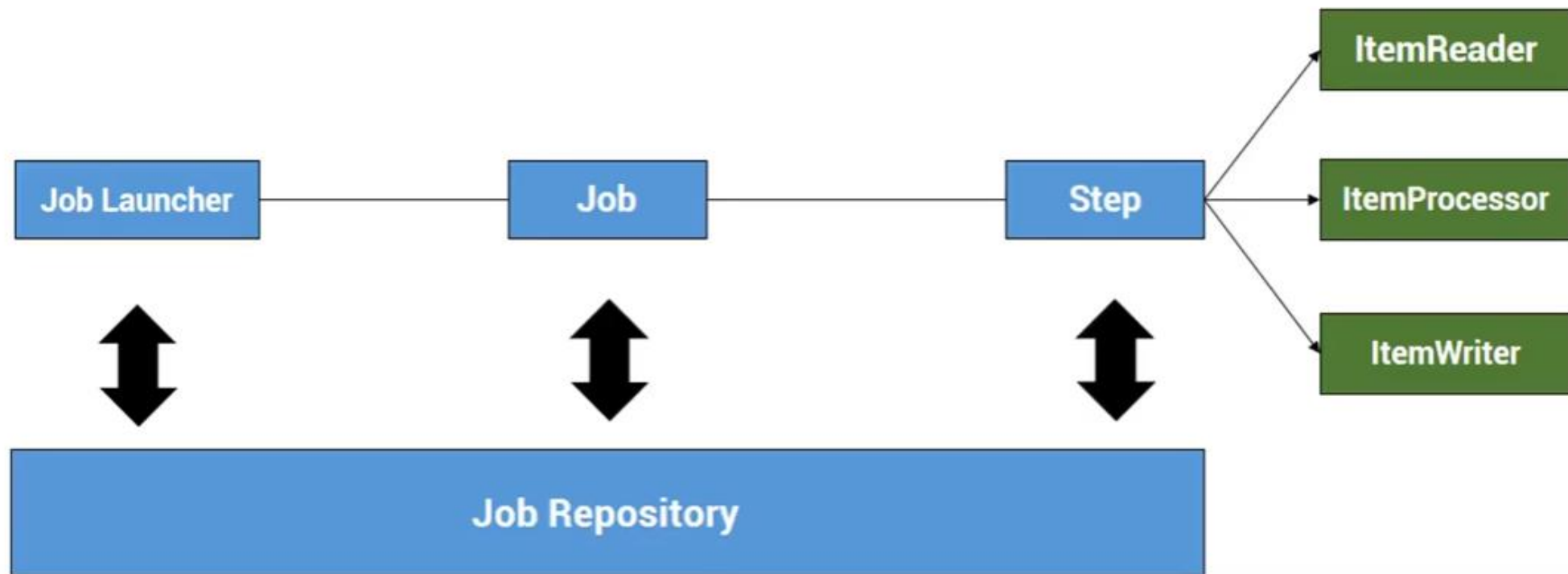
Que es Spring Batch

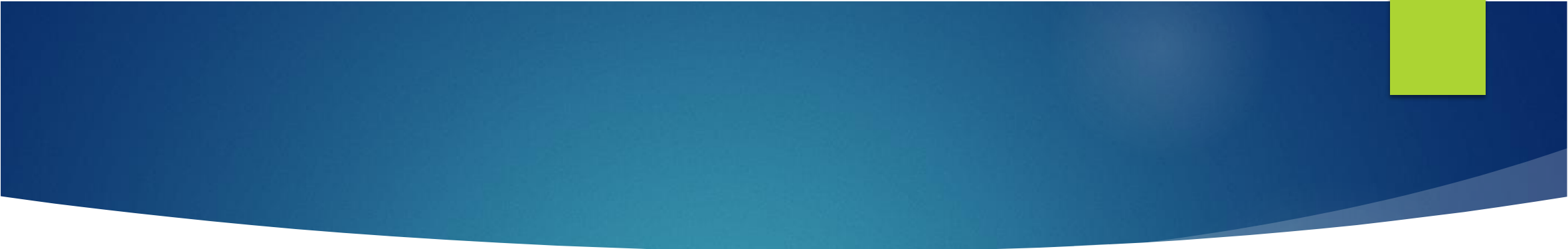
- ▶ Es un modulo de Spring framework totalmente enfocado en los procesos batch, el cual adicionalmente proporciona una gran cantidad de componentes que intentan dar soporte a las diferentes necesidades que suelen surgir al momento de construir este tipo de programas

Funcionalidades de Spring Batch


- ▶ Administración de la transacción
- ▶ Procesamiento en “Chunk”
- ▶ Declaración de I/O
- ▶ Iniciar/Detener/Reiniciar
- ▶ Volver a intentar/Omitir
- ▶ Administración web(Spring Cloud Data Flow)

Componentes de Spring Batch



- 
- ▶ **Job Launcher:** Es el componente encargado de lanzar los procesos subministrando los parámetros de entradas solicitados por cada uno de los procesos, cabe resaltar que un job launcher debe tener como mínimo un Job.
 - ▶ **Job:** es la representación de un proceso, un proceso a su vez es el contenido de uno muchos pasos(Step)
 - ▶ **Step:** es un elemento independiente dentro de un Job que representa una de las fases de las que esta compuesto dicho proceso.
 - ▶ Es necesario que un Job tenga un Step, pero que podemos tener muchos Jobs y Steps
 - ▶ Los Step pueden estar compuestos de 3 elementos un ItemReader , ItemProccesor, ItemWriter
 - ▶ **ItemReader:** Es el responsable de leer los datos de una fuente de datos ya sea una base de datos un fichero, una cola de mensaje entre otros,
 - ▶ **ItemProccesor:** es el responsable de tratar la información obtenida, este paso es opcional
 - ▶ **Itemwriter:** es el responsable de guardar la información leída por el reader o tratada por el proccesor, si existe un reader debe existir al menos un writer. Puede ser una base de datos un archivo o cualquier otro elemento donde se pueda depositar o enviar la información obtenida y procesada.

Veremos ejemplo de volcar archivos desde csv a base de datos, para ello creamos un nuevo proyecto con las siguientes dependencias



spring initializr

Project
☒ Maven Project ☐ Gradle Project

Language
☒ Java ☐ Kotlin ☐ Groovy

Spring Boot
☐ 3.0.0 (SNAPSHOT) ☐ 3.0.0 (M1) ☐ 2.7.0 (SNAPSHOT) ☐ 2.7.0 (M1)
☐ 2.6.4 (SNAPSHOT) ☒ 2.6.3 ☐ 2.5.10 (SNAPSHOT) ☐ 2.5.9

Project Metadata

Group

Artifact

Name

Description

Package name

Packaging ☒ Jar ☐ War

Java ☐ 17 ☐ 11 ☒ 8

Dependencies ADD DEPENDENCIES... CTRL + B

JDBC API SQL
Database Connectivity API that defines how a client may connect and query a database.

MySQL Driver SQL
MySQL JDBC and R2DBC driver.

Spring Batch I/O
Batch applications with transactions, retry/skip and chunk based processing.

GENERATE CTRL + G

EXPLORE CTRL + SPACE

SHARE...

Creamos la base de datos

localhost/phpmyadmin/index.php?route=/server/databases&server=1

Servidor: 127.0.0.1

Bases de datos SQL Estado actual Cuentas de usuarios Exportar





Bases de datos

Crear base de datos ?

demo_batch utf8mb4_general_ci Crear

Filtros

Que contengan la palabra:

Base de datos	Cotejamiento	Acción
<input type="checkbox"/> api_directorio	utf8mb4_general_ci	 Seleccionar privilegios
<input type="checkbox"/> curso_laravel	utf8mb4_general_ci	 Seleccionar privilegios
<input type="checkbox"/> db_apiproductos	utf8mb4_general_ci	 Seleccionar privilegios
<input type="checkbox"/> db_apirest	utf8mb4_general_ci	 Seleccionar privilegios

localhost/phpmyadmin/index.php?route=/database/structure&server=1&db=demo_batch

Servidor: 127.0.0.1 » Base de datos: demo_batch » Tabla: persona

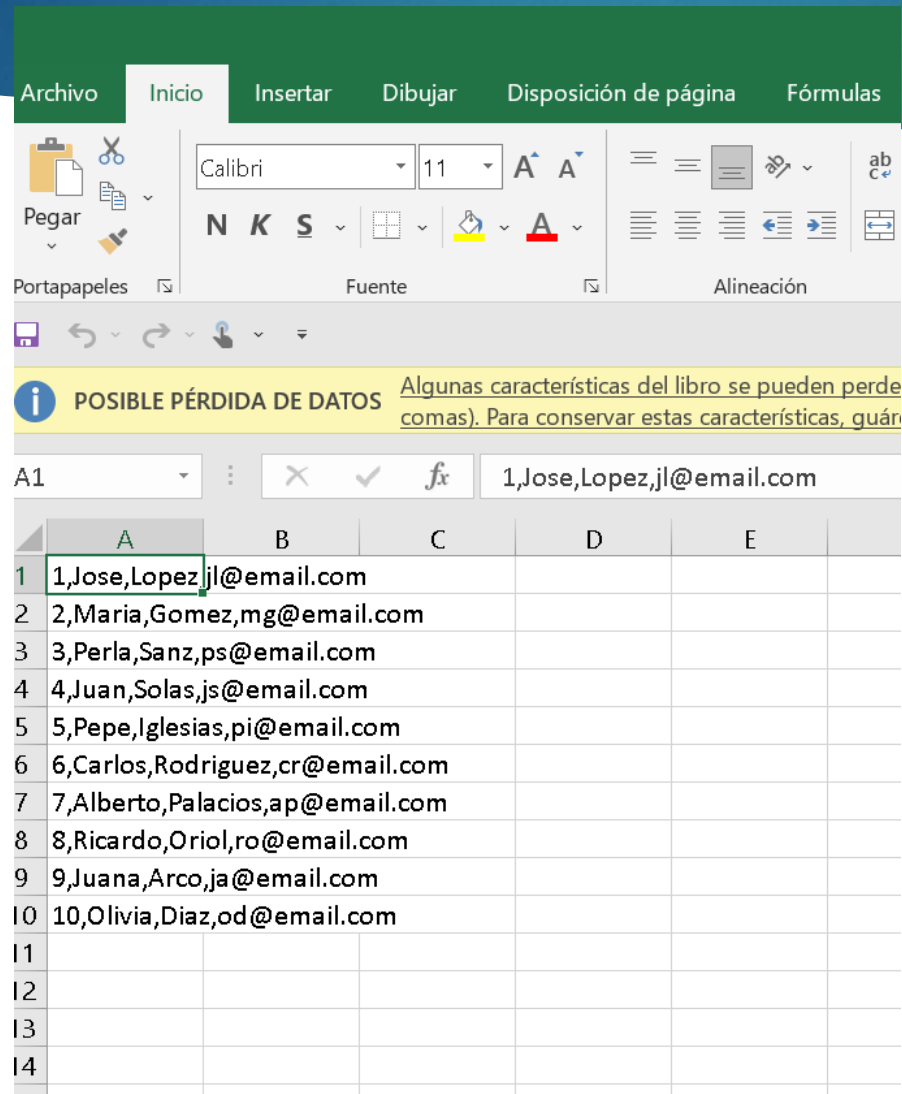
Examinar Estructura SQL Buscar Insertar Exportar Importar















Nombre de la tabla: persona Agregar 1 columna(s) Continuar

Nombre	Tipo	Longitud/Valores	Predeterminado	Cotejam
id	BIGINT		Ninguno	
nombre	VARCHAR	50	Ninguno	
apellido	VARCHAR	50	Ninguno	
email	VARCHAR	50	Ninguno	

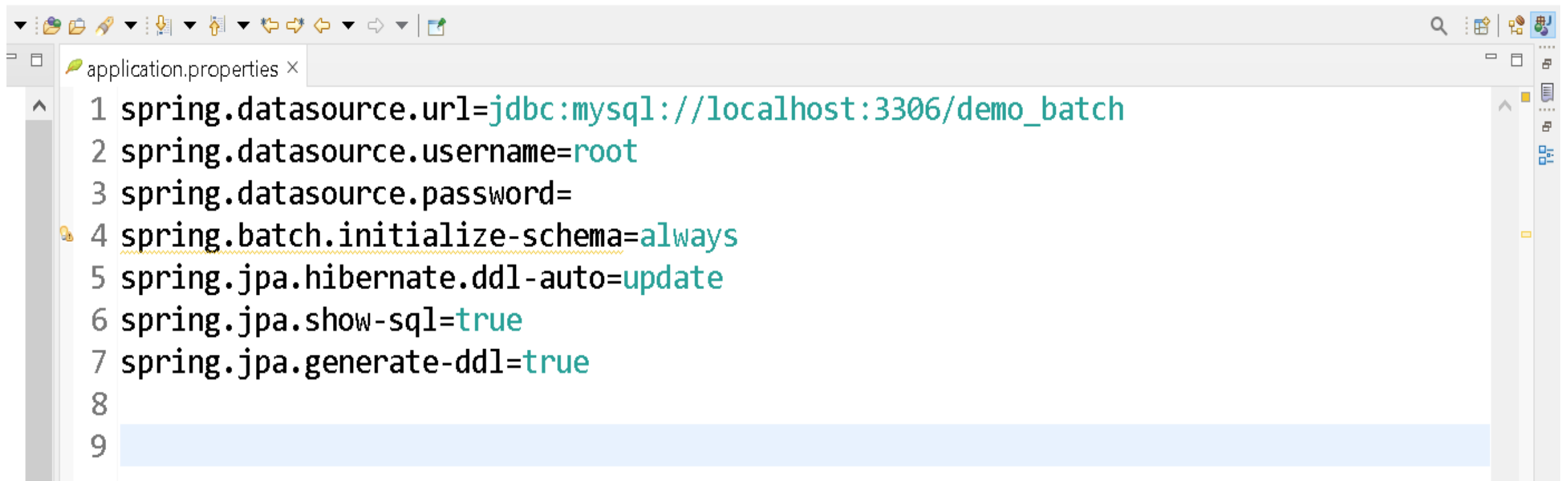
Estructura

Agregamos el siguiente archivo csv al proyecto en resources



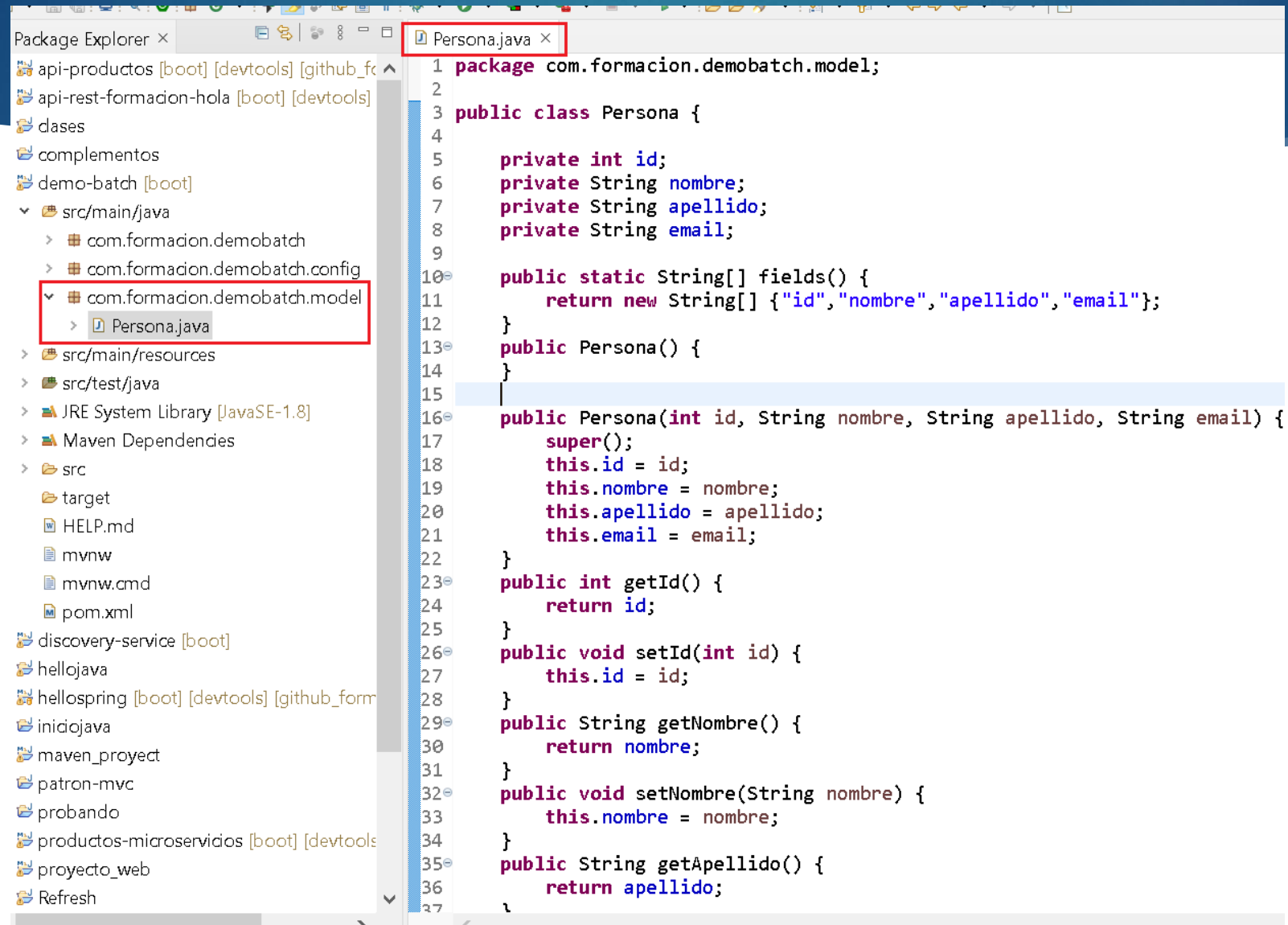
- ▼  demo-batch [boot]
 - >  src/main/java
 - ▼  src/main/resources
 -  application.properties
 -  persona.csv
 - >  src/test/java
 - >  JRE System Library [JavaSE-1.8]
 - >  Maven Dependencies
 - >  src
 -  target
 -  HELP.md
 -  mvnw
 -  mvnw.cmd
 -  pom.xml

Configuramos el application properties

A screenshot of an IDE window titled 'application.properties'. The window shows a list of configuration properties for a Spring application. The properties are numbered 1 through 9. Line 4 has a yellow squiggly underline under 'spring.batch.initialize-schema'. The IDE interface includes a toolbar at the top with various icons for file operations and a search icon. The right side of the window shows a sidebar with icons for project structure, search, and other IDE features.

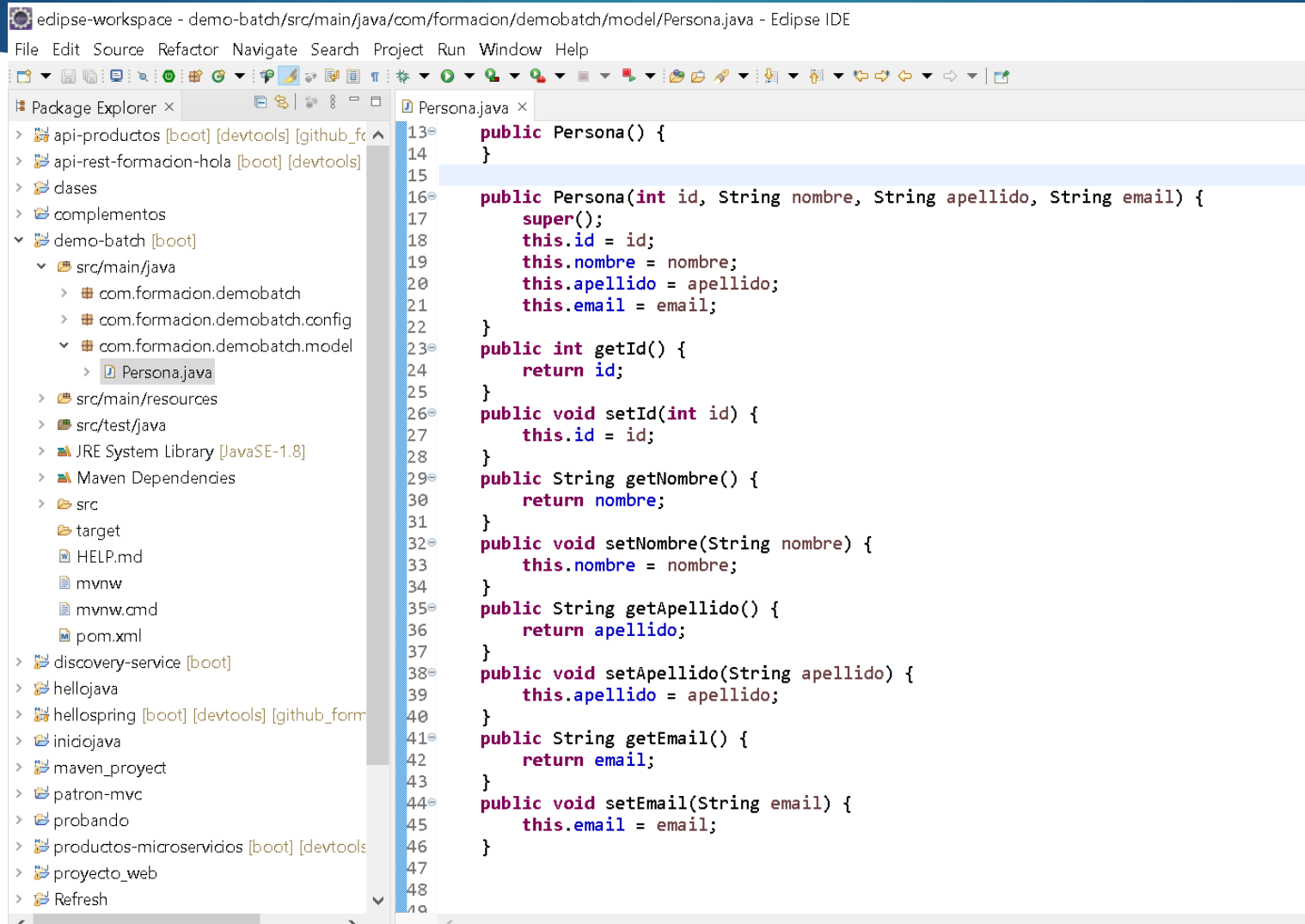
```
1 spring.datasource.url=jdbc:mysql://localhost:3306/demo_batch
2 spring.datasource.username=root
3 spring.datasource.password=
4 spring.batch.initialize-schema=always
5 spring.jpa.hibernate.ddl-auto=update
6 spring.jpa.show-sql=true
7 spring.jpa.generate-ddl=true
8
9
```

Creo un modelo persona



```
1 package com.formacion.demobatch.model;
2
3 public class Persona {
4
5     private int id;
6     private String nombre;
7     private String apellido;
8     private String email;
9
10    public static String[] fields() {
11        return new String[] { "id", "nombre", "apellido", "email" };
12    }
13    public Persona() {
14    }
15
16    public Persona(int id, String nombre, String apellido, String email) {
17        super();
18        this.id = id;
19        this.nombre = nombre;
20        this.apellido = apellido;
21        this.email = email;
22    }
23    public int getId() {
24        return id;
25    }
26    public void setId(int id) {
27        this.id = id;
28    }
29    public String getNombre() {
30        return nombre;
31    }
32    public void setNombre(String nombre) {
33        this.nombre = nombre;
34    }
35    public String getApellido() {
36        return apellido;
37    }
38 }
```

Con los constructores setters y getters



edipse-workspace - demo-batch/src/main/java/com/formacion/demobatch/model/Persona.java - Edipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer x

- api-productos [boot] [devtools] [github_fc]
- api-rest-formacion-hola [boot] [devtools]
- dases
- complementos
- demo-batch [boot]
 - src/main/java
 - com.formacion.demobatch
 - com.formacion.demobatch.config
 - com.formacion.demobatch.model
 - Persona.java
 - src/main/resources
 - src/test/java
 - JRE System Library [JavaSE-1.8]
 - Maven Dependencies
 - src
 - target
 - HELP.md
 - mvnw
 - mvnw.cmd
 - pom.xml
- discovery-service [boot]
- hellojava
- hellospring [boot] [devtools] [github_form]
- inidjava
- maven_proyect
- patron-mvc
- probando
- productos-microservidos [boot] [devtools]
- proyecto_web
- Refresh

Persona.java x

```
13 public Persona() {
14 }
15
16 public Persona(int id, String nombre, String apellido, String email) {
17     super();
18     this.id = id;
19     this.nombre = nombre;
20     this.apellido = apellido;
21     this.email = email;
22 }
23 public int getId() {
24     return id;
25 }
26 public void setId(int id) {
27     this.id = id;
28 }
29 public String getNombre() {
30     return nombre;
31 }
32 public void setNombre(String nombre) {
33     this.nombre = nombre;
34 }
35 public String getApellido() {
36     return apellido;
37 }
38 public void setApellido(String apellido) {
39     this.apellido = apellido;
40 }
41 public String getEmail() {
42     return email;
43 }
44 public void setEmail(String email) {
45     this.email = email;
46 }
47
48
49
```

Creo el siguiente archivo de configuración

edipse-workspace - demo-batch/src/main/java/com/formacion/demobatch/config/PersonaBatchConfiguration.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer ×

- > api-productos [boot] [devtools] [github_fr
- > api-rest-formacion-hola [boot] [devtools]
- > clases
- > complementos
- > demo-batch [boot]
 - > src/main/java
 - > com.formacion.demobatch
 - > **com.formacion.demobatch.config**
 - > **PersonaBatchConfiguration.java**
 - > com.formacion.demobatch.model
 - > src/main/resources
 - > src/test/java
 - > JRE System Library [JavaSE-1.8]
 - > Maven Dependencies
 - > src
 - target
 - HELP.md
 - mvnw
 - mvnw.cmd
 - pom.xml
 - > discovery-service [boot]
 - > hellojava
 - > hellospring [boot] [devtools] [github_form
 - > iniciojava
 - > maven_proyect
 - > patron-mvc

*PersonaBatchConfiguration.java ×

```
1 package com.formacion.demobatch.config;
2
3 import javax.sql.DataSource;
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23 @Configuration
24 @EnableBatchProcessing
25 public class PersonaBatchConfiguration {
26
27     @Autowired
28     private StepBuilderFactory stepBuilderFactory;
29
30     @Autowired
31     private JobBuilderFactory jobBuilderFactory;
32
33     @Autowired
34     private DataSource dataSource;
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
```

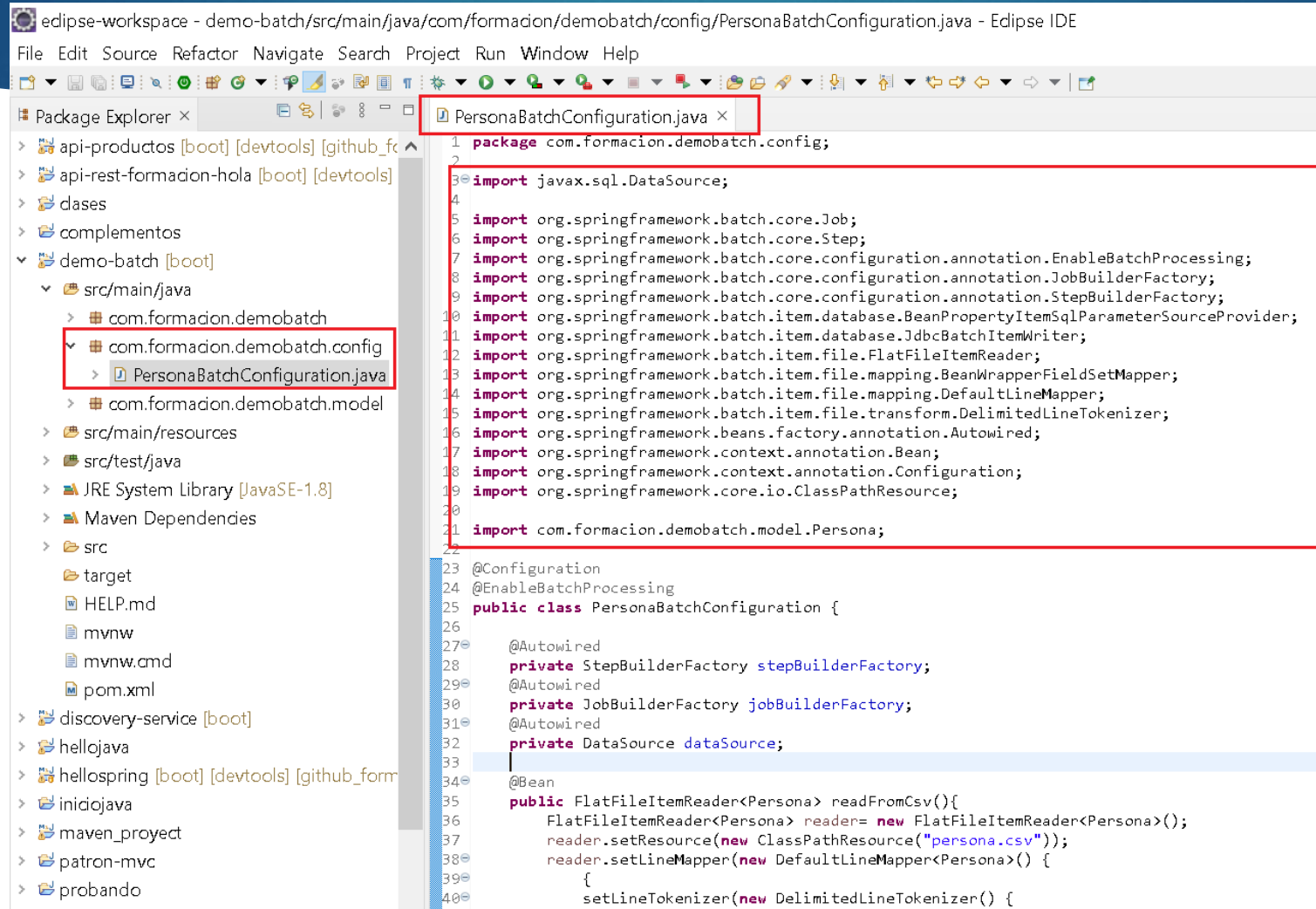

Package Explorer x

- > api-productos [boot] [devtools] [github_fc]
- > api-rest-formacion-hola [boot] [devtools]
- > clases
- > complementos
- ▼ demo-batch [boot]
 - ▼ src/main/java
 - > com.formacion.demobatch
 - ▼ com.formacion.demobatch.config
 - > PersonaBatchConfiguration.java
 - > com.formacion.demobatch.model
 - > src/main/resources
 - > src/test/java
 - > JRE System Library [JavaSE-1.8]
 - > Maven Dependencies
 - > src
 - target
 - HELP.md
 - mvnw
 - mvnw.cmd
 - pom.xml
 - > discovery-service [boot]
 - > hellojava
 - > hellospring [boot] [devtools] [github_form]
 - > iniciojava
 - > maven_proyect
 - > patron-mvc
 - > probando
 - > productos-microservicios [boot] [devtools]
 - > proyecto_web
 - > Refresh

*PersonaBatchConfiguration.java x

```
37 reader.setResource(new ClassPathResource("persona.csv"));
38 reader.setLineMapper(new DefaultLineMapper<Persona>() {
39     {
40         setLineTokenizer(new DelimitedLineTokenizer() {
41             {
42                 setNames(Persona.fields());
43             }
44         });
45     }
46
47     setFieldSetMapper(new BeanWrapperFieldSetMapper<Persona>() {
48         {
49             setTargetType(Persona.class);
50         }
51     });
52 }
53 });
54
55 return reader;
56 }
57
58 @Bean
59 public JdbcBatchItemWriter<Persona> writerIntoDB(){
60     JdbcBatchItemWriter<Persona> writer = new JdbcBatchItemWriter<Persona>();
61     writer.setDataSource(dataSource);
62     writer.setSql("insert into persona(id,nombre,apellido,email) values(:id,:nombre,:apellido,:email);");
63     writer.setItemSqlParameterSourceProvider(new BeanPropertyItemSqlParameterSourceProvider<Persona>());
64     return writer;
65 }
66
67
68 @Bean
69 public Step step() {
70     return stepBuilderFactory.get("step").<Persona,Persona>chunk(10)
71         .reader(readFromCsv()).writer(writerIntoDB()).build();
72 }
73
74 @Bean
75 public Job job() {
76     return jobBuilderFactory.get("job").flow(step()).end().build();
77 }
78
79 }
```

Los imports



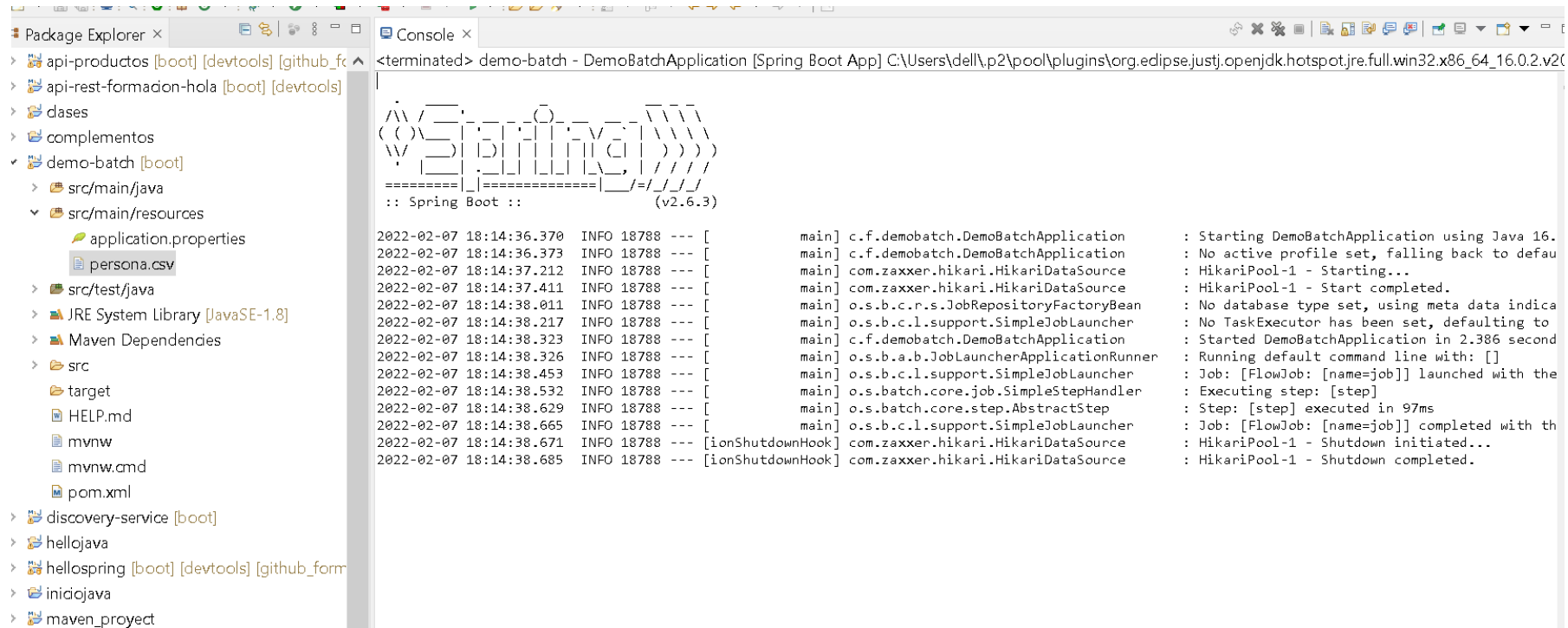
The screenshot shows the Eclipse IDE interface. The Package Explorer on the left displays a project structure with a package `com.formacion.demobatch.config` containing the file `PersonaBatchConfiguration.java`. The main editor shows the code for `PersonaBatchConfiguration.java`. The imports section, highlighted with a red box, includes:

```
1 package com.formacion.demobatch.config;
2
3 import javax.sql.DataSource;
4
5 import org.springframework.batch.core.Job;
6 import org.springframework.batch.core.Step;
7 import org.springframework.batch.core.configuration.annotation.EnableBatchProcessing;
8 import org.springframework.batch.core.configuration.annotation.JobBuilderFactory;
9 import org.springframework.batch.core.configuration.annotation.StepBuilderFactory;
10 import org.springframework.batch.item.database.BeanPropertyItemSqlParameterSourceProvider;
11 import org.springframework.batch.item.database.JdbcBatchItemWriter;
12 import org.springframework.batch.item.file.FlatFileItemReader;
13 import org.springframework.batch.item.file.mapping.BeanWrapperFieldSetMapper;
14 import org.springframework.batch.item.file.mapping.DefaultLineMapper;
15 import org.springframework.batch.item.file.transform.DelimitedLineTokenizer;
16 import org.springframework.beans.factory.annotation.Autowired;
17 import org.springframework.context.annotation.Bean;
18 import org.springframework.context.annotation.Configuration;
19 import org.springframework.core.io.ClassPathResource;
20
21 import com.formacion.demobatch.model.Persona;
```

The code continues with annotations and class definition:

```
22
23 @Configuration
24 @EnableBatchProcessing
25 public class PersonaBatchConfiguration {
26
27     @Autowired
28     private StepBuilderFactory stepBuilderFactory;
29
30     @Autowired
31     private JobBuilderFactory jobBuilderFactory;
32
33     @Autowired
34     private DataSource dataSource;
35
36     @Bean
37     public FlatFileItemReader<Persona> readFromCsv(){
38         FlatFileItemReader<Persona> reader= new FlatFileItemReader<Persona>();
39         reader.setResource(new ClassPathResource("persona.csv"));
40         reader.setLineMapper(new DefaultLineMapper<Persona>() {
41             {
42                 setLineTokenizer(new DelimitedLineTokenizer() {
```

Compilamos



Debemos ver el resultado datos desde el csv a base de datos

MyAdmin

Servidor: 127.0.0.1 » Base de datos: demo_batch » Tabla: persona

Examinar Estructura SQL Buscar Insertar Exportar

⚠ La selección actual no contiene una columna única. La edición de la grilla y los enlaces de copia no están disponibles.

✓ Mostrando filas 0 - 9 (total de 10, La consulta tardó 0,0008 segundos.)

`SELECT * FROM `persona``

☐ Perfilando [[Editar en línea](#)] [[Editar](#)] [[Explicar SQL](#)] [[Crear código PHP](#)] [[Actualizar](#)]


☐ Mostrar todo | Número de filas: 25 ▼ | Filtrar filas:

+ Opciones

id	nombre	apellido	email
1	Jose	Lopez	jl@email.com
2	Maria	Gomez	mg@email.com
3	Perla	Sanz	ps@email.com
4	Juan	Solas	js@email.com
5	Pepe	Iglesias	pi@email.com
6	Carlos	Rodriguez	cr@email.com
7	Alberto	Palacios	ap@email.com
8	Ricardo	Oriol	ro@email.com
9	Juana	Arco	ja@email.com
10	Olivia	Diaz	od@email.com

☐ Mostrar todo | Número de filas: 25 ▼ | Filtrar filas:

Ahora otro ejemplo generar un csv a partir de base de datos



Project
☒ Maven Project ☐ Gradle Project

Language
☒ Java ☐ Kotlin ☐ Groovy



Spring Boot
☐ 3.0.0 (SNAPSHOT) ☐ 3.0.0 (M1) ☐ 2.7.0 (SNAPSHOT) ☐ 2.7.0 (M1)
☐ 2.6.4 (SNAPSHOT) ☒ 2.6.3 ☐ 2.5.10 (SNAPSHOT) ☐ 2.5.9

Project Metadata
Group
Artifact
Name
Description
Package name
Packaging ☒ Jar ☐ War
Java ☐ 17 ☐ 11 ☒ 8

Dependencies ADD DEPENDENCIES... CTRL + B

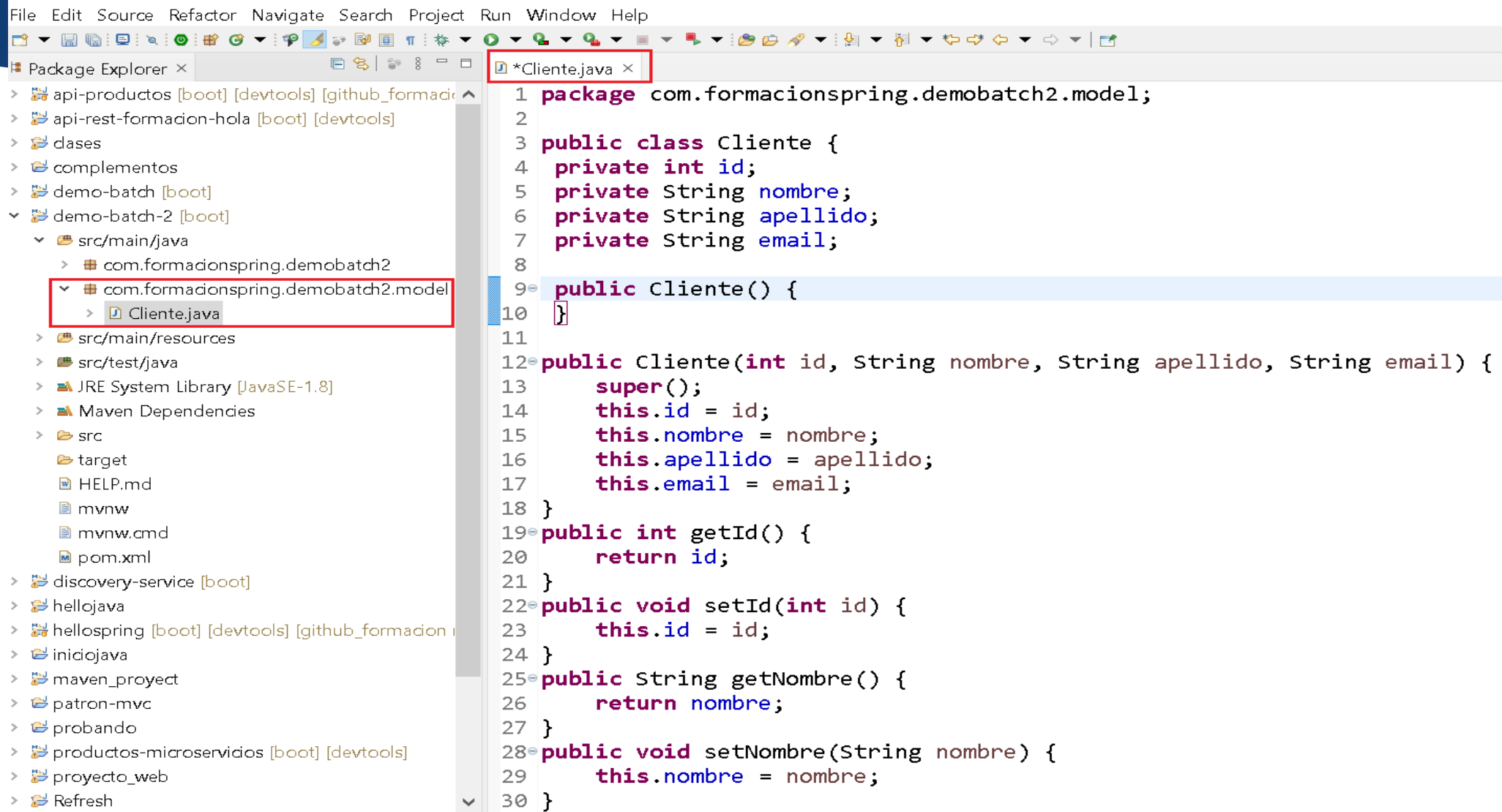
Spring Batch I/O
Batch applications with transactions, retry/skip and chunk based processing.

MySQL Driver SQL
MySQL JDBC and R2DBC driver.



GENERATE CTRL + G EXPLORE CTRL + SPACE SHARE...

Creamos el siguiente modelo



The screenshot shows an IDE with the Package Explorer on the left and the Java editor on the right. In the Package Explorer, the package `com.formacionspring.demobatch2.model` is selected, and the file `Cliente.java` is being created. The Java editor displays the following code:

```
1 package com.formacionspring.demobatch2.model;
2
3 public class Cliente {
4     private int id;
5     private String nombre;
6     private String apellido;
7     private String email;
8
9     public Cliente() {
10    }
11
12    public Cliente(int id, String nombre, String apellido, String email) {
13        super();
14        this.id = id;
15        this.nombre = nombre;
16        this.apellido = apellido;
17        this.email = email;
18    }
19    public int getId() {
20        return id;
21    }
22    public void setId(int id) {
23        this.id = id;
24    }
25    public String getNombre() {
26        return nombre;
27    }
28    public void setNombre(String nombre) {
29        this.nombre = nombre;
30    }
```

edipse-workspace - demo-batch-2/src/main/java/com/formacionspring/demobatch2/model/Cliente.java - Edipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

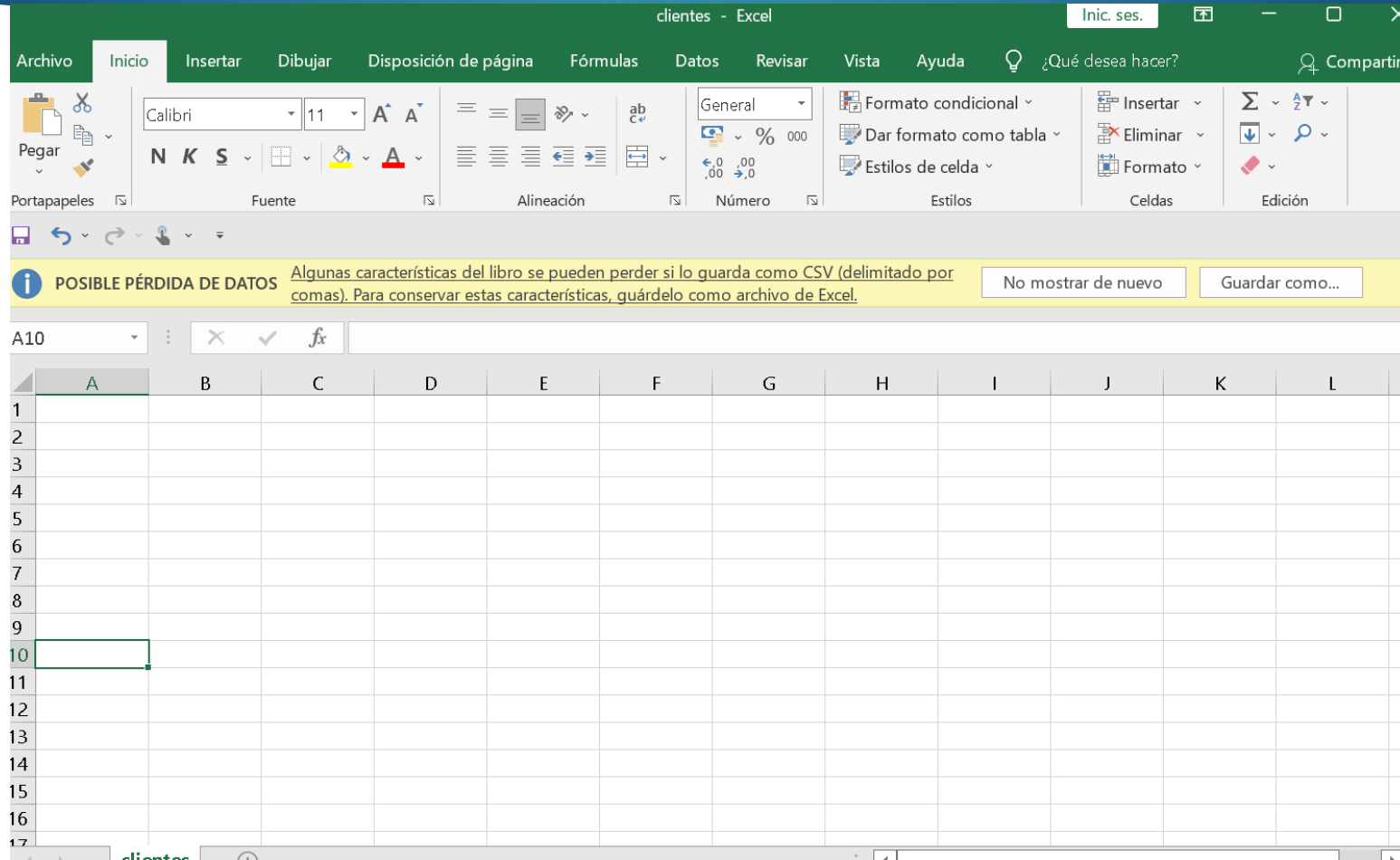
Package Explorer

- > api-productos [boot] [devtools] [github_formacion]
- > api-rest-formacion-hola [boot] [devtools]
- > clases
- > complementos
- > demo-batch [boot]
- ▼ demo-batch-2 [boot]
 - ▼ src/main/java
 - > com.formacionspring.demobatch2
 - ▼ com.formacionspring.demobatch2.model
 - > Cliente.java
 - > src/main/resources
 - > src/test/java
 - > JRE System Library [JavaSE-1.8]
 - > Maven Dependencies
 - > src
 - target
 - HELP.md
 - mvnw
 - mvnw.cmd
 - pom.xml
 - > discovery-service [boot]
 - > hellojava
 - > hellospring [boot] [devtools] [github_formacion]
 - > iniciojava
 - > maven_proyect
 - > patron-mvc
 - > probando
 - > productos-microservicios [boot] [devtools]
 - > proyecto_web
 - > Refresh

*Cliente.java

```
18 }
19 public int getId() {
20     return id;
21 }
22 public void setId(int id) {
23     this.id = id;
24 }
25 public String getNombre() {
26     return nombre;
27 }
28 public void setNombre(String nombre) {
29     this.nombre = nombre;
30 }
31 public String getApellido() {
32     return apellido;
33 }
34 public void setApellido(String apellido) {
35     this.apellido = apellido;
36 }
37 public String getEmail() {
38     return email;
39 }
40 public void setEmail(String email) {
41     this.email = email;
42 }
43
44
45
46 }
47
```

Ubicamos un archivo csv vacio en el directorio que deseamos



Archivo de configuración

edipse-workspace - demo-batch-2/src/main/java/com/formacionspring/demobatch2/config/BatchConfiguration.java - Edipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer ×

- complementos
- demo-batch [boot]
- demo-batch-2 [boot]
 - src/main/java
 - com.formacionspring.demobatch2
 - com.formacionspring.demobatch2.config
 - BatchConfiguration.java
 - com.formacionspring.demobatch2.model
 - src/main/resources
 - src/test/java
 - JRE System Library [JavaSE-1.8]
 - Maven Dependencies
 - src
 - target
 - HELP.md
 - mvnw
 - mvnw.cmd
 - pom.xml
 - discovery-service [boot]
 - hellojava
 - hellospring [boot] [devtools] [github_formacion]
 - inidiojava
 - maven_proyect
 - patron-mvc
 - probando
 - productos-microservicios [boot] [devtools]
 - proyecto_web
 - Refresh
 - repaso
 - segundo [boot] [devtools]

application.properties BatchConfiguration.java ×

```
1 package com.formacionspring.demobatch2.config;
2
3 import java.sql.ResultSet;
4
5
6 @Configuration
7 @EnableBatchProcessing
8 public class BatchConfiguration {
9
10     @Autowired
11     private StepBuilderFactory stepBuilderFactory;
12
13     @Autowired
14     private JobBuilderFactory jobBuilderFactory;
15
16     @Autowired
17     private DataSource dataSource;
18
19     @Bean
20     public JdbcCursorItemReader<Cliente> reader(){
21         JdbcCursorItemReader<Cliente> reader = new JdbcCursorItemReader<Cliente>();
22         reader.setDataSource(dataSource);
23         reader.setSql("select id,nombre,apellido,email from persona");
24         reader.setRowMapper(new RowMapper<Cliente>() {
25
26             @Override
27             public Cliente mapRow(ResultSet rs, int rowNum) throws SQLException {
28                 Cliente s = new Cliente();
29                 s.setId(rs.getInt("id"));
30                 s.setNombre(rs.getString("nombre"));
31                 s.setApellido(rs.getString("apellido"));
32                 s.setEmail(rs.getString("email"));
33                 return s;
34             }
35         });
36     }
37 }
```

Package Explorer ×

- > complementos
- > demo-batch [boot]
- ▼ demo-batch-2 [boot]
 - ▼ src/main/java
 - > com.formacionspring.demobatch2
 - ▼ com.formacionspring.demobatch2.config
 - > BatchConfiguration.java
 - > com.formacionspring.demobatch2.model
 - > src/main/resources
 - > src/test/java
 - > JRE System Library [JavaSE-1.8]
 - > Maven Dependencies
 - > src
 - target
 - HELP.md
 - mvnw
 - mvnw.cmd
 - pom.xml
 - > discovery-service [boot]
 - > hellojava
 - > hellospring [boot] [devtools] [github_formacion]
 - > iniciojava
 - > maven_proyect
 - > patron-mvc
 - > probando
 - > productos-microservicios [boot] [devtools]
 - > proyecto_web
 - > Refresh
 - > repaso
 - > segundo [boot] [devtools]

application.properties

BatchConfiguration.java ×

```
36 @Autowired
37 private DataSource dataSource;
38
39 @Bean
40 public JdbcCursorItemReader<Cliente> reader(){
41     JdbcCursorItemReader<Cliente> reader = new JdbcCursorItemReader<Cliente>();
42     reader.setDataSource(dataSource);
43     reader.setSql("select id,nombre,apellido,email from persona");
44     reader.setRowMapper(new RowMapper<Cliente>() {
45
46         @Override
47         public Cliente mapRow(ResultSet rs, int rowNum) throws SQLException {
48             Cliente s = new Cliente();
49             s.setId(rs.getInt("id"));
50             s.setNombre(rs.getString("nombre"));
51             s.setApellido(rs.getString("apellido"));
52             s.setEmail(rs.getString("email"));
53             return s;
54         }
55     });
56
57     return reader;
58 }
59
60 @Bean
61 public FlatFileItemWriter<Cliente> writer(){
62     FlatFileItemWriter<Cliente> writer= new FlatFileItemWriter<Cliente>();
63     writer.setResource(new FileSystemResource("C:\\Users\\dell\\Desktop\\Curso_Enero\\clientes.csv"));
64     DelimitedLineAggregator<Cliente> aggregator= new DelimitedLineAggregator<>();
65     BeanWrapperFieldExtractor<Cliente> fieldExtractor= new BeanWrapperFieldExtractor<>();
66     fieldExtractor.setNames(new String[] { "id", "nombre", "apellido", "email" });
67     aggregator.setFieldExtractor(fieldExtractor);
68     writer.setLineAggregator(aggregator);
69     return writer;
70 }
71
72 }
```

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer

- > complementos
- > demo-batch [boot]
- ▼ demo-batch-2 [boot]
 - ▼ src/main/java
 - > com.formacionspring.demobatch2
 - ▼ com.formacionspring.demobatch2.config
 - > BatchConfiguration.java
 - > com.formacionspring.demobatch2.model
 - > src/main/resources
 - > src/test/java
 - > JRE System Library [JavaSE-1.8]
 - > Maven Dependencies
 - > src
 - target
 - HELP.md
 - mvnw
 - mvnw.cmd
 - pom.xml
 - > discovery-service [boot]
 - > hellojava
 - > hellospring [boot] [devtools] [github_formacion]
 - > iniciojava
 - > maven_proyect
 - > patron-mvc
 - > probando
 - > productos-microservicios [boot] [devtools]
 - > proyecto_web
 - > Refresh
 - > repaso
 - > segundo [boot] [devtools]

application.properties

BatchConfiguration.java

```
48 Cliente s = new Cliente();
49 s.setId(rs.getInt("id"));
50 s.setNombre(rs.getString("nombre"));
51 s.setApellido(rs.getString("apellido"));
52 s.setEmail(rs.getString("email"));
53 return s;
54 }
55
56 });
57
58 return reader;
59 }
60
61 @Bean
62 public FlatFileItemWriter<Cliente> writer(){
63     FlatFileItemWriter<Cliente> writer= new FlatFileItemWriter<Cliente>();
64     writer.setResource(new FileSystemResource("C:\\Users\\dell\\Desktop\\Curso_Enero\\clientes.csv"));
65     DelimitedLineAggregator<Cliente> aggregator= new DelimitedLineAggregator<>();
66     BeanWrapperFieldExtractor<Cliente> fieldExtractor= new BeanWrapperFieldExtractor<>();
67     fieldExtractor.setNames(new String[] {"id", "nombre", "apellido", "email"});
68     aggregator.setFieldExtractor(fieldExtractor);
69     writer.setLineAggregator(aggregator);
70     return writer;
71 }
72
73 @Bean
74 public Step executeStep() {
75     return stepBuilderFactory.get("executeStep").<Cliente, Cliente>chunk(10).reader(reader()).writer(writer()).build();
76 }
77
78 @Bean
79 public Job processJob() {
80     return jobBuilderFactory.get("processJob").incrementer(new RunIdIncrementer()).flow(executeStep()).end().build();
81 }
82
83 }
84
```

edipse-workspace - demo-batch-2/src/main/java/com/formacionspring/demobatch2/config/BatchConfiguration.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help



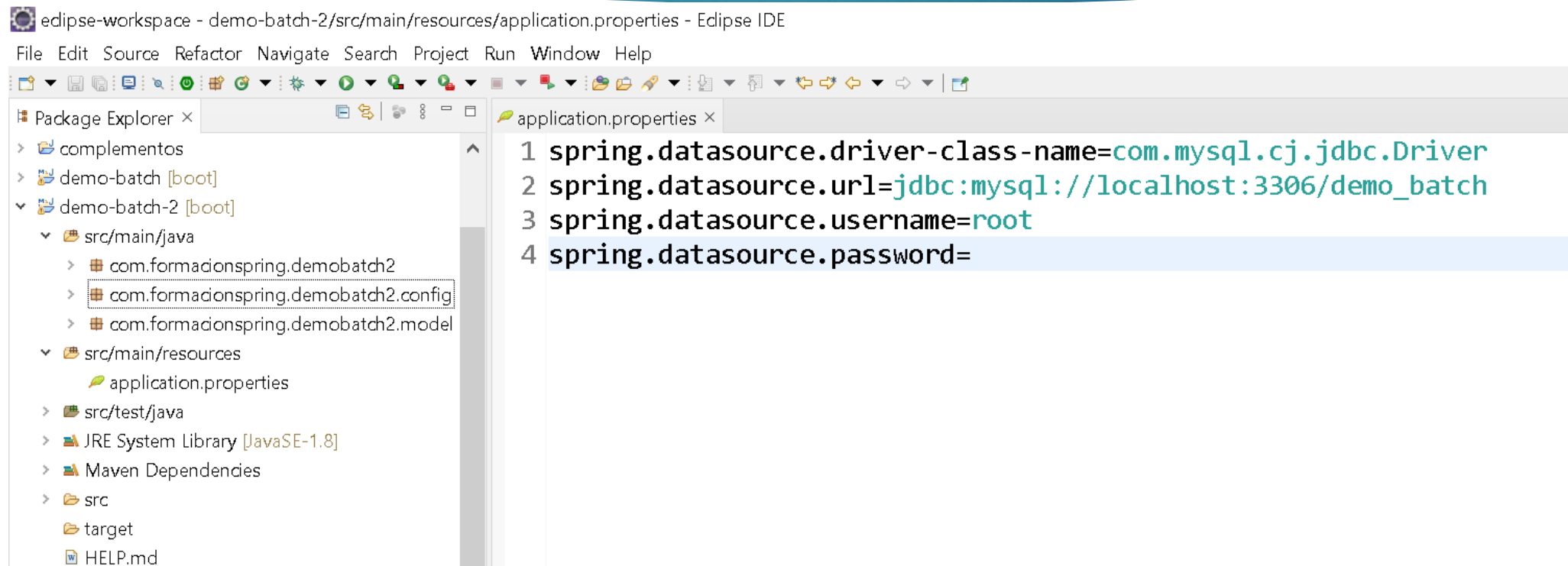
Package Explorer ×

- > complementos
- > demo-batch [boot]
- ▼ demo-batch-2 [boot]
 - ▼ src/main/java
 - > com.formacionspring.demobatch2
 - ▼ com.formacionspring.demobatch2.config
 - > BatchConfiguration.java
 - > com.formacionspring.demobatch2.model
 - > src/main/resources
 - > src/test/java
 - > JRE System Library [JavaSE-1.8]
 - > Maven Dependencies
 - > src
 - target
 - HELP.md
 - mvnw
 - mvnw.cmd
 - pom.xml
 - > discovery-service [boot]
 - > hellojava
 - > hellospring [boot] [devtools] [github_formacion]
 - > iniciojava
 - > maven_proyect
 - > patron-mvc
 - > probando
 - > productos-microservicios [boot] [devtools]
 - > proyecto_web
 - > Refresh
 - > repaso
 - > segundo [boot] [devtools]

application.properties BatchConfiguration.java ×

```
1 package com.formacionspring.demobatch2.config;
2
3 import java.sql.ResultSet;
4 import java.sql.SQLException;
5
6 import javax.sql.DataSource;
7
8 import org.springframework.batch.core.Job;
9 import org.springframework.batch.core.Step;
10 import org.springframework.batch.core.configuration.annotation.EnableBatchProcessing;
11 import org.springframework.batch.core.configuration.annotation.JobBuilderFactory;
12 import org.springframework.batch.core.configuration.annotation.StepBuilderFactory;
13 import org.springframework.batch.core.launch.support.RunIdIncrementer;
14 import org.springframework.batch.item.database.JdbcCursorItemReader;
15 import org.springframework.batch.item.file.FlatFileItemWriter;
16 import org.springframework.batch.item.file.transform.BeanWrapperFieldExtractor;
17 import org.springframework.batch.item.file.transform.DelimitedLineAggregator;
18 import org.springframework.beans.factory.annotation.Autowired;
19 import org.springframework.context.annotation.Bean;
20 import org.springframework.context.annotation.Configuration;
21 import org.springframework.core.io.FileSystemResource;
22 import org.springframework.jdbc.core.RowMapper;
23
24 import com.formacionspring.demobatch2.model.Cliente;
25
26 @Configuration
27 @EnableBatchProcessing
28 public class BatchConfiguration {
29
30     @Autowired
31     private StepBuilderFactory stepBuilderFactory;
32
33     @Autowired
34     private JobBuilderFactory jobBuilderFactory;
35
36     @Autowired
37     private DataSource dataSource;
```

Configuramos el application.properties a la base de datos que ya tenemos



localhost/phpmyadmin/index.php?route=/sql&server=1&db=demo_batch&table=persona&pos=0

phpMyAdmin

Reciente

Favoritas

Nueva

+ api_directorio

+ curso_laravel

+ db

- demo_batch

Nueva

+ persona

+ information_schema

+ laravel_passaport

+ mysql

+ performance_schema

+ phpmyadmin

+ prueba

+ python_database

+ spring_producto

+ test

Servidor: 127.0.0.1 » Base de datos: demo_batch » Tabla: persona

Examinar

Estructura

SQL

Buscar

Insertar

Exportar

Importar

Privilegios

⚠ La selección actual no contiene una columna única. La edición de la grilla y los enlaces de copiado, eliminación y edición no están disponibles.

✓ Mostrando filas 0 - 9 (total de 10, La consulta tardó 0,0008 segundos.)

```
SELECT * FROM `persona`
```

☐ Perfilando [[Editar en línea](#)] [[Editar](#)] [[Explicar SQL](#)] [[Crear código PHP](#)] [[Actualizar](#)]

☐ Mostrar todo | Número de filas: 25 ▾ | Filtrar filas:

+ Opciones

id	nombre	apellido	email
1	Jose	Lopez	jl@email.com
2	Maria	Gomez	mg@email.com
3	Perla	Sanz	ps@email.com
4	Juan	Solas	js@email.com
5	Pepe	Iglesias	pi@email.com
6	Carlos	Rodriguez	cr@email.com
7	Alberto	Palacios	ap@email.com
8	Ricardo	Oriol	ro@email.com
9	Juana	Arco	ja@email.com
10	Olivia	Diaz	od@email.com

100

[illegible]

