JUAN JESÚS ÁVILA MOTA

Academia Java Xideral

Examen Semana03




PROGRAM WITH JUNIT

Proyectos - JUnitProyect/src/com/examen/TestExamen.java - Eclipse IDE

File Edit Source Refactor Navigate Search Project Run Window Help

Package Explorer | JUnit

Finished after 0.227 seconds

Runs: 5/5 Errors: 0 Failures: 0

TestExamen [Runner: JUnit 5] (0.026 s)

Failure Trace

TestExamen.java | Empleado.java

```
45      void testPagoHorasExtra() {
46          int horasExtra = 0;
47          double pagoHorasExtra = 0;
48          Empleado e = new Empleado("José","Cajero",100,10);
49          if(e.trabajoExtra())
50          horasExtra = e.getHorasTrabajo() - 8;
51          pagoHorasExtra = horasExtra * (e.getSueldoHora() * 2);
52          assertEquals(pagoHorasExtra,e.pagoHorasExtra());
53      }
54      //Prueba JUnit para valida que los nombres no contengan números
55      @DisplayName("Validar Nombres")
56      @Test
57      void testValidaNombre() {
58          Empleado e = new Empleado("José","Cajero",100,10);
59          boolean cN = e.getNombre().matches(".*[0-9].*");
60          assertEquals(cN,e.validaNombre());
61      }
62
63
64  }
65
```

Problems | Javadoc | Declaration | Console | Terminal

<terminated> TestExamen [JUnit] C:\Program Files\Java\jre1.8.0_202\bin\javaw.exe (5 dic. 2022 20:06:03 – 20:06:05) [pid: 18308]

Writable | Smart Insert | 61 : 6 : 2113

Escribe aquí para buscar

78% 14°C 08:06 p. m. 05/12/2022

TDD was used in this program, that is, the JUnit tests were programmed first and later the class was created with its attributes and methods, where it is validated if the employee worked overtime and what was the payment for these hours which are paid to the double

SCRUM

11 important points have been established to implement SCRUM, this methodology is not a recipe that must be followed specifically, it is implemented according to the purpose and better adaptability of the project and the team

1.- It is necessary to establish the person responsible for the product, Product Owner. It has the vision of what is going to be done, produced, achieved and the rewards of the project.

2.- Select a good work team, the people who are part of the team are capable of understanding the vision of the product owner and turning the vision into reality, the team to be a small team of 3 to 9 people.

3.- Select the Scrum Master, who will be in charge of integrating the team into the SCRUM methodology and eliminating what harms the team.

4.- Create the product log, a list of what has to be done in order to turn the vision into reality, applying emphasis on what is a priority.

5.- Refine the log, together with the work team, estimate the effort that is carried out in each of the listed activities and verify if they can be carried out correctly with the resource that is available.

6.- Plan the Sprint, this is the first meeting, the product owner, Spring Master and the team, the sprints have a fixed duration of less than one month, based on the log it is planned how long the sprint can take, not this subject to change.

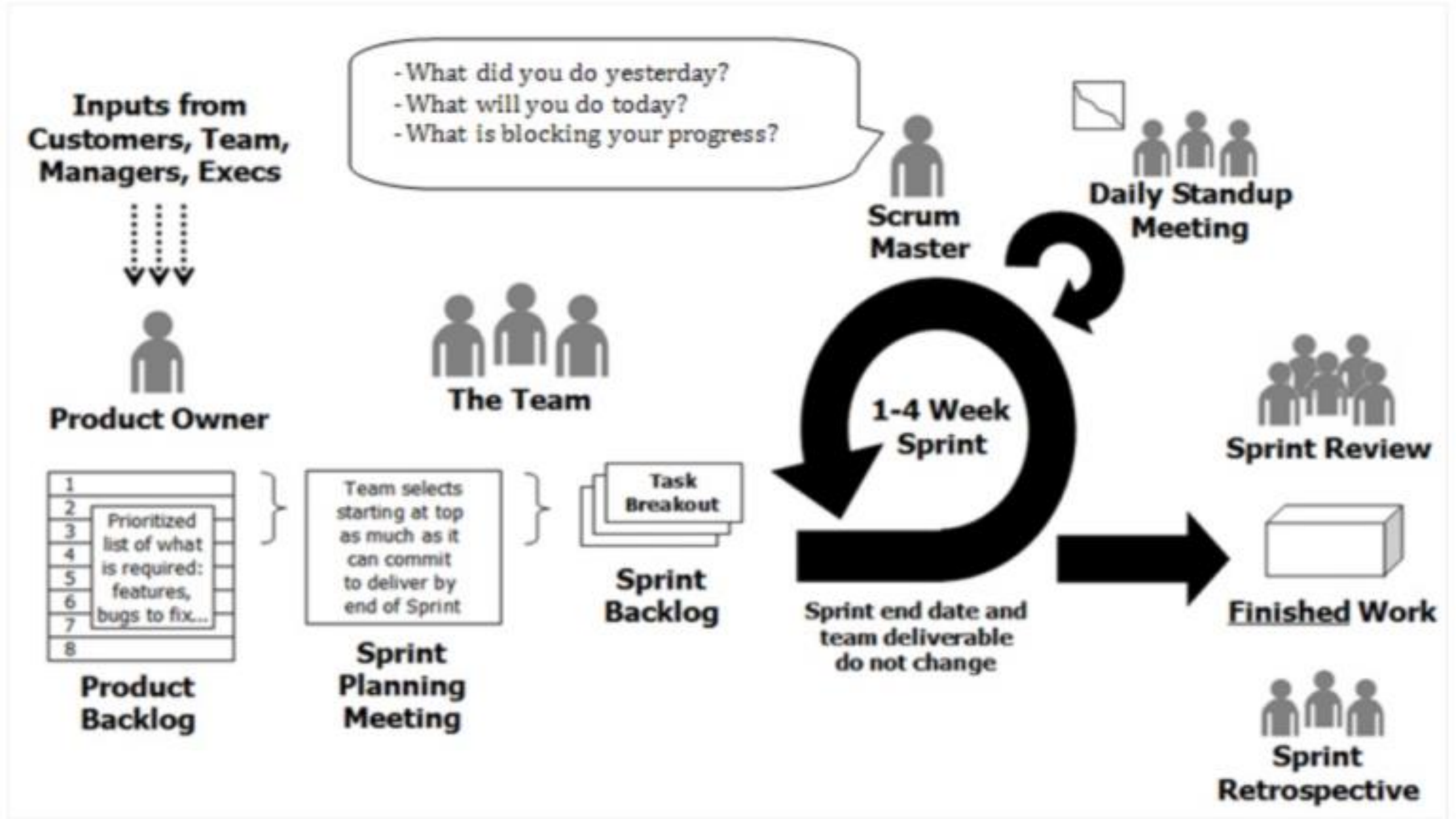
7.- Make the work visible, create a Scrum table with three columns, pending, in process, finished, posticks represent the processes that are carried out and change columns depending on the status of the element.

8.- Daily Scrum, meeting no longer than 15 minutes, the team answers three fundamental questions:
What did you do yesterday to help the team finish the sprint?
What will you do today to help the team finish the sprint?
Are any obstacles preventing you or the team from meeting the sprint goal?

9.- Review of the Sprint, what was done during the Sprint is shown, everyone involved can attend, from the client to the entire Scrum team, which team demonstrates what is considered 100% finished

10.-Feedback from the Sprint, where you have an item from the finished log and ask what can be done to improve? What was done well? What can be done faster? Taking into account the maturity of the team and the trust that exists between them

11.- Start the Sprint again, with possible improvements, due to the experience acquired by the team


Inputs from Customers, Team, Managers, Execs
-What did you do yesterday?
-What will you do today?
-What is blocking your progress?
Scrum Master
Daily Standup Meeting
Product Owner
The Team
1-4 Week Sprint
Sprint Review
1
2
3
4
5
6
7
8
Prioritized list of what is required: features, bugs to fix...
Team selects starting at top as much as it can commit to deliver by end of Sprint
Task Breakout
Sprint Backlog
Finished Work
Sprint end date and team deliverable do not change
Product Backlog
Sprint Planning Meeting
Sprint Retrospective




CRUD WEB SERVLETS

Deportistas

Agregar Deportista

Nombre	Apellido Paterno	Apellido Materno	Deporte	Acción
Juan	Avila	Mota	Soccer	Update | Delete
Carolina	Lopez	Chavez	Soccer	Update | Delete
Jesus	Perez	Alvarez	Basquetball	Update | Delete
Erik	Estrada	Sauza	Soccer	Update | Delete
Julio	Montes	Moreno	Soccer	Update | Delete
Erika	Delgado	Manzur	Soccer	Update | Delete
Juan Jesus	Mota	Avila	Soccer	Update | Delete
nombre	aPaterno	aMaterno	deporte	Update | Delete

The program shown is a CRUD which uses servlets and jsp, as well as MVC with an architecture in which the model takes the role of interacting with the database through JDBC.
The database was created in MYSQL, where athlete data is stored.
Dependencies were pulled in using Maven

CRUD WEB SPRING AND HIBERNATE

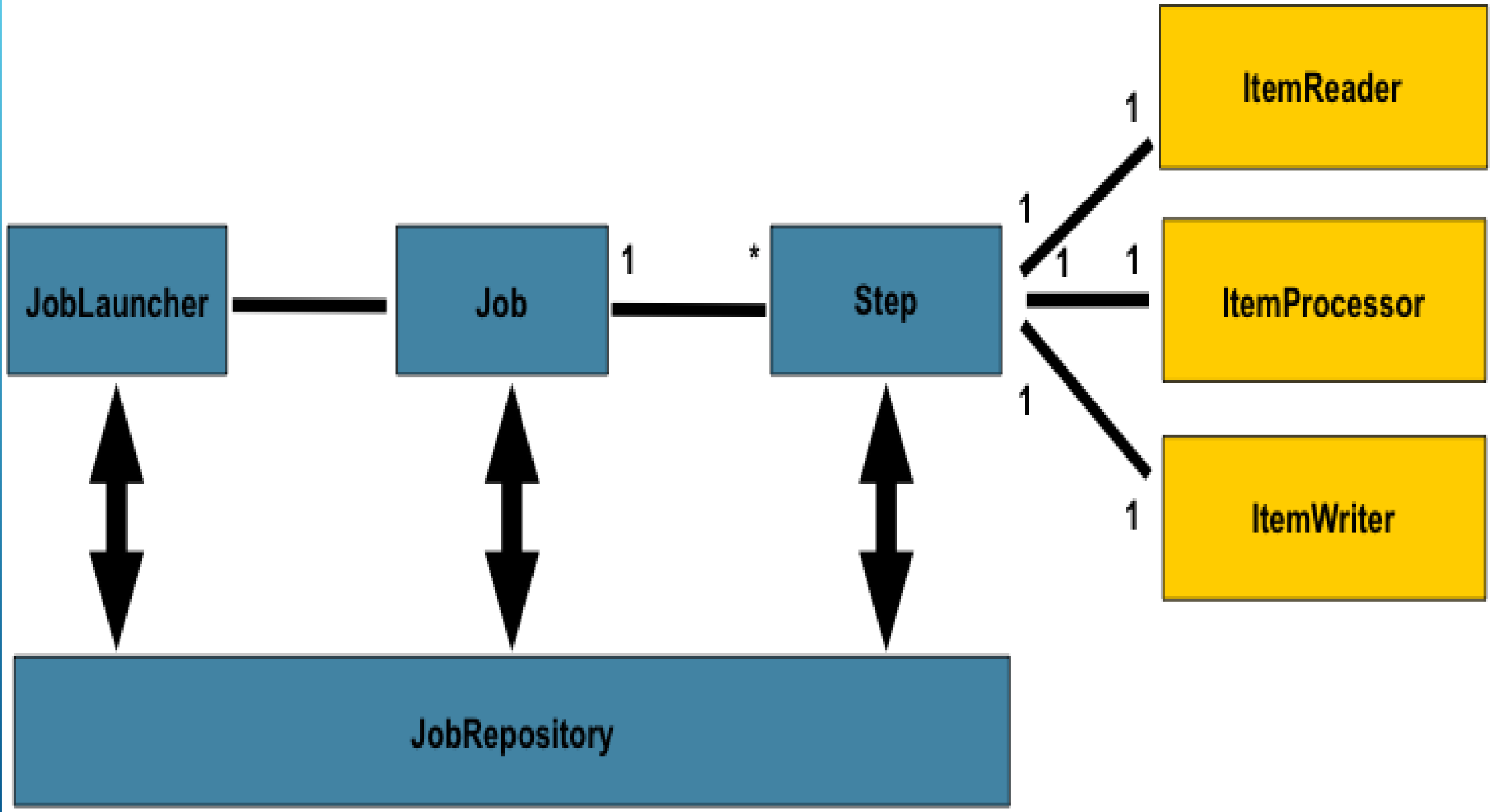
Deportista

Agregar Deportist

Nombre	Apellido paterno	Apellido materno	Deporte	Acciones
Juan	Avila	Mota	Baseball	Actualizar | Eliminar
Jesus	Perez	Alvarez	Basquetball	Actualizar | Eliminar
Carolina	Lopez	Chavez	Soccer	Actualizar | Eliminar
Miguel	Lopez	Hernandez	Soccer	Actualizar | Eliminar

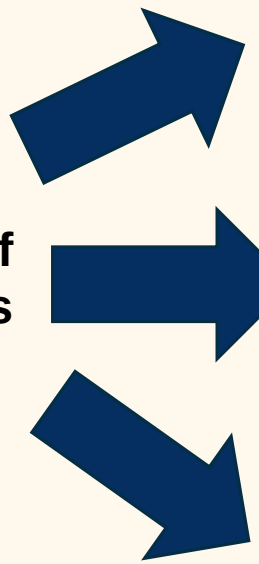
The program is a crud using Spring and Hibernate, these frameworks, minimize the lines of code obtained in the previous exercise, seeing the fundamental purpose of any framework.
Being Hibernate capable of mapping the data extracted from the database and sharing this information through the Java files that make up the project.
The dependencies were pulled in via Maven.

SPRING BATCH


ItemReader
1
1
JobLauncher
Job
1
*
Step
1
1
ItemProcessor
1
1
ItemWriter
JobRepository

Spring batch is thinking about memory processing. This repository is used mainly for writing, although it is also consulted to check if a file has already been processed previously.

A job is a block of work and is made up of one or several steps. Once all these steps have been carried out, the job is considered complete.



ItemReader: Handles reading from batch processing. This reading can be, for example, from a database; Or it could also be from a message broker or a csv, xml, json, etc. file.

ItemProccessor: is in charge of transforming previously read items. This transformation, in addition to including changes in the format, can include data filtering or business logic.

ItemWriter: este elemento es lo opuesto al itemReader. Se encarga de la escritura de los ítems. Esta puede ser inserciones en una base de datos, en un fichero csv, en un broker de mensajes, etc.