



## **Arquitectura De Software**

### **Realizado por:**

Esteban A Villa Berrio

Stefania García

## **Laboratorio 3 – Arquitectura**

### **Presentado a:**

Andrés Armando Sanchez Martin

Noviembre del 2022

## Marco conceptual

El objetivo de este laboratorio es aprender sobre distintas herramientas y realizar una práctica para su uso. A lo largo de este documento se mostrará el procedimiento que se tuvo que realizar para este laboratorio y también las lecciones aprendidas.

A continuación, se explicarán las herramientas que se utilizaron con sus características para el desarrollo.

1. GitHub: Es un sitio “social coding” que permite subir repositorios de código para almacenarlo en el sistema de versión de controles de Git, se puede colaborar en proyectos de código, y el sistema es código abierto por defecto, lo que significa que cualquier persona en el mundo puede encontrar tu código, usarlo, aprender de él, y mejorarlo.
  - ¿Para qué se utilizó? Esta herramienta fue de gran utilidad donde al haber creado el repositorio en el primer laboratorio permitió subir los archivos necesarios del desarrollo del laboratorio 3
  - Procedimiento: Primero se ingresó al github, se ingresó al repositorio creado llamado personapi-dotnet y se subió una carpeta con lo necesario en el desarrollo del laboratorio 3
2. Apache NetBeans IDE 15: Proporciona editores, asistentes y plantillas para ayudarlo a crear aplicaciones en Java, PHP y muchos otros lenguajes. Apache NetBeans se puede instalar en todos los sistemas operativos compatibles con Java.
  - ¿Para qué se utilizó? Esta herramienta fue la base de todo el laboratorio ya que permitió crear el código de solución y la conexión a la base de datos
  - Procedimiento: Se realizó la descarga de la página directa de NetBeans, como ya se tenía el jdk sólo fue necesario instalar el Apache y se abre el entorno de desarrollo después de ejecutar
3. MariaDB: Es una de las bases de datos relacionales de código abierto más populares del mundo y está disponible en los repositorios estándar de todas las principales distribuciones de Linux
  - ¿Para qué se utilizó? Esta herramienta permitió almacenar información y llevar a cabo los proyectos de manera eficiente
  - Procedimiento: Se realizó la descarga, al ejecutar se verifica si incluye Heidi SQL, se pone una contraseña y confirmación de contraseña para cualquier conexión necesaria y se deja la configuración predeterminada

4. Heidi SQL: Es un software libre y tiene como objetivo que es de fácil aprendizaje. Permite ver y editar los datos, estructuras, computadoras que ejecutan uno de los sistemas de base de datos MariaDB, MySQL, Microsoft SQL, PostgreSQL y SQLite.
  - ¿Para qué se utilizó? Esta herramienta fue de gran utilidad para conectarse con las bases de datos de MariaDB
  - Procedimiento: Primero se realizó la descarga por medio de Google Chrome, se ejecuta es muy importante que la opción de que se asocien los archivos de SQL con Heidi SQL y así ya está el espacio de trabajo disponible para utilizar
5. Github Desktop: GitHub Desktop es una aplicación que te habilita para interactuar con GitHub utilizando una GUI en vez de la línea de comandos o de un buscador web. GitHub Desktop fomenta que tú y tu equipo colaboren utilizando las mejoras prácticas con Git y GitHub. Puedes utilizar GitHub Desktop para completar la mayoría de los comandos de Git desde tu computadora de escritorio con confirmaciones visuales para los cambios.
  - ¿Para que se utilizó? Esta herramienta permitió subir los cambios que se realizaban en el repositorio generado desde la web por medio de una carpeta con la clonación de este
  - Procedimiento: Se seleccionó el repositorio creado, se clonó en una carpeta en el computador de manera local, y al momento de guardar un archivo en dicha carpeta se utilizaba el desktop para dar commit y push
6. MySQL Worbench: Es una herramienta visual unificada para arquitectos de bases de datos, desarrolladores y administradores de bases de datos. MySQL proporciona modelado de datos, desarrollo de SQL y herramientas de administración integrales para la configuración del servidor, la administración de usuarios, la copia de seguridad y mucho más
  - ¿Para qué se utilizó? Esta herramienta fue de gran utilidad para crear una nueva conexión a una base de datos para realizar las tablas correspondientes
  - Procedimiento: Primero se realizó la descarga de la aplicación por medio de Google Chrome, al ejecutar la descarga se seleccionan la forma completa y al terminar de instalarlo se abre el espacio de trabajo
7. MongoDB Compass: Es una poderosa GUI para consultar, agregar y analizar sus datos de MongoDB en un entorno visual
  - ¿Para qué se utilizó? Esta herramienta permitió crear la base de datos con su respectiva información para facilitar la conexión al proyecto
  - Procedimiento: Primero se realizó la descarga, luego se inicio sesión, y se escogió el enlace de conexión

## Procedimiento

Para el desarrollo de este laboratorio se realizaron los siguientes pasos:

1. Se creó un repositorio en github llamado personapi-dotnet de manera pública que se puede encontrar en la url <https://github.com/DianaStefaniaGarcia/personapi-dotnet>

personapi-dotnet

Public

Laboratorio 1

2. Se buscó en internet la página Maven Repository Central:  
<https://mvnrepository.com/repos/central>
3. Buscamos MariaDB y seleccionamos la que dice MariaDB Java Client, en la pestaña org.mariadb.jdbc donde se escogerá la versión 3.0.9

**MVN REPOSITORY** Search for groups, artifacts, categories Search

Indexed Artifacts (30.6M)

Popular Categories

- Testing Frameworks
- Android Packages
- Logging Frameworks
- Java Specifications
- JSON Libraries
- Core Utilities
- JVM Languages
- Mocking
- Language Runtime
- Web Assets

Home » org.mariadb.jdbc » mariadb-java-client

**MariaDB Java Client**  
JDBC driver for MariaDB and MySQL

License: **LGPL 2.1**

Categories: **JDBC Drivers**

Tags: database, sql, jdbc, driver, mariadb, client, mysql

Ranking: #614 in MvnRepository (See Top Artifacts)  
#7 in JDBC Drivers

Used By: 668 artifacts

Central (96) Clojars (1) Spring Plugins (1) Redhat GA (6) ICM (1)

Version	Vulnerabilities	Repository	Usages	Date
3.0.9		Central	2	Nov 08, 2022
3.0.8		Central	25	Sep 20, 2022
3.0.7		Central	27	Aug 04, 2022
3.0.6		Central	29	Jun 29, 2022

4. Luego, se verán las distintas características del Maven, Gradle y demás

MVN REPOSITORY

Search for groups, artifacts, categories

Search

Indexed Artifacts (30.6M)



Popular Categories

Testing Frameworks

Android Packages

Logging Frameworks

Java Specifications

JSON Libraries

Core Utilities

JVM Languages

Mocking

Language Runtime

Web Assets

Annotation Libraries

Logging Bridges

HTTP Clients

Home » org.mariadb.jdbc » mariadb-java-client » 3.0.9



MariaDB Java Client » 3.0.9

JDBC driver for MariaDB and MySQL

License

Categories

Tags

Organization

HomePage

Date

Files

Repositories

Ranking

Used By

LGPL 2.1

JDBC Drivers

database sql jdbc driver mariadb client mysql

mariadb.com

https://mariadb.com/kb/en/mariadb/about-mariadb-connector-j/

Nov 08, 2022

pom (17 KB) jar (577 KB) View All

Central

#614 in MvnRepository (See Top Artifacts)

#7 in JDBC Drivers

668 artifacts

Maven

Gradle

Gradle (Short)

Gradle (Kotlin)

SBT

Ivy

Grape

Leiningen

Buildr

<!-- https://mvnrepository.com/artifact/org.mariadb.jdbc/mariadb-java-client -->

<dependency>

<groupId>org.mariadb.jdbc</groupId>

<artifactId>mariadb-java-client</artifactId>

<version>3.0.9</version>

</dependency>

Home » org.mariadb.jdbc » mariadb-java-client » 3.0.9



MariaDB Java Client » 3.0.9

JDBC driver for MariaDB and MySQL

License

Categories

Tags

Organization

HomePage

Date

Files

Repositories

Ranking

Used By

LGPL 2.1

JDBC Drivers

database sql jdbc driver mariadb client mysql

mariadb.com

https://mariadb.com/kb/en/mariadb/about-mariadb-connector-j/

Nov 08, 2022

pom (17 KB) jar (577 KB) View All

Central

#614 in MvnRepository (See Top Artifacts)

#7 in JDBC Drivers

668 artifacts

Maven

Gradle

Gradle (Short)

Gradle (Kotlin)

SBT

Ivy

Grape

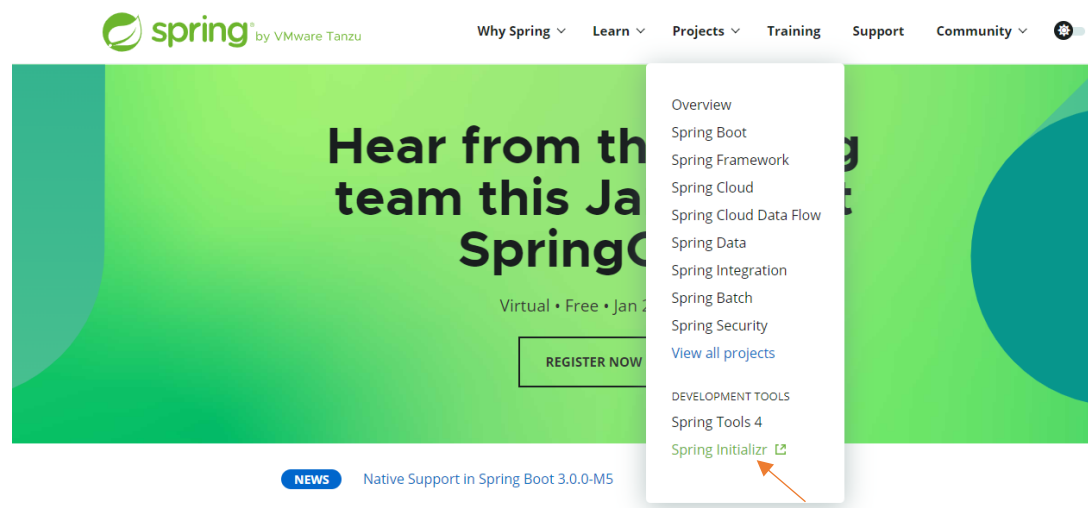
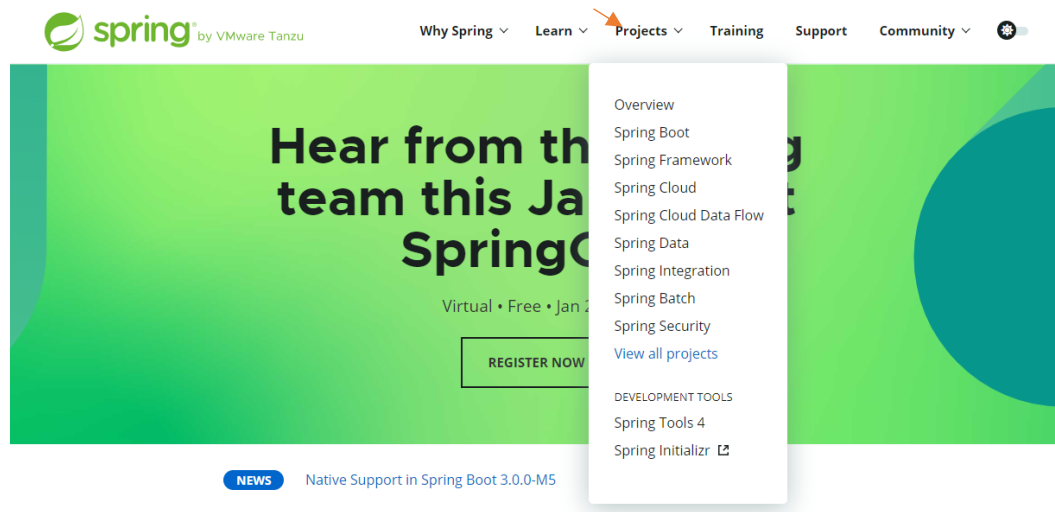
Leiningen

Buildr

// https://mvnrepository.com/artifact/org.mariadb.jdbc/mariadb-java-client

implementation group: 'org.mariadb.jdbc', name: 'mariadb-java-client', version: '3.0.9'

5. Después, nos dirigimos a la página web de spring: <https://spring.io/>, luego en la pestaña projects damos clic en Spring Initializr



6. Aparecerá una nueva ventana donde se creará un nuevo proyecto con las siguientes características: Proyecto gradle, lenguaje Java, la opción por defecto de Spring Boot, el grupo es co.edu.javeriana.as.boot.spring.personapp, el artefacto y el nombre es personapp-spring-boot, tal como se ve a continuación

The screenshot shows the Spring Initializr web form. On the left, there's a sidebar with a hamburger menu and social media icons. The main form is divided into sections: **Project** (with radio buttons for Gradle Project, Maven Project, and Groovy), **Language** (with radio buttons for Java, Kotlin, and Groovy), and **Spring Boot** (with radio buttons for 3.0.0 (SNAPSHOT), 3.0.0 (RC2), 2.7.6 (SNAPSHOT), 2.7.5, 2.6.14 (SNAPSHOT), and 2.6.13). Below these is the **Project Metadata** section with input fields for Group (co.edu.javeriana.as.boot.spring.personapp), Artifact (personapp-spring-boot), Name (personapp-spring-boot), Description (Demo project for Spring Boot), and Package name (co.edu.javeriana.as.boot.spring.personapp.personapp-spring). There are also checkboxes for Packaging (Jar, War) and Java version (19, 17, 11, 8). On the right, there's a section for dependencies with the text "No dependency selected". At the bottom, there are three buttons: GENERATE (with a keyboard shortcut CTRL + G), EXPLORE (with a keyboard shortcut CTRL + SPACE), and SHARE...

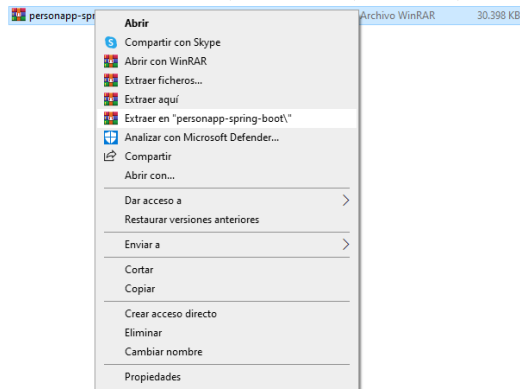
7. Se da clic en agregar dependencias y se utilizan: Spring for GraphQL, Spring Data JPA, Spring web, MariaDB Driver, Lombok, Spring Data MongoDB

The screenshot shows the Spring Initializr web form with dependencies added. The form is the same as in the previous screenshot, but with additional dependencies listed on the right side. The dependencies are: **Spring Web** (WEB), **Spring for GraphQL** (WEB), **Spring Data JPA** (SQL), **MariaDB Driver** (SQL), **Lombok** (DEVELOPER TOOLS), and **Spring Data MongoDB** (NO SQL). Each dependency has a brief description of its functionality.

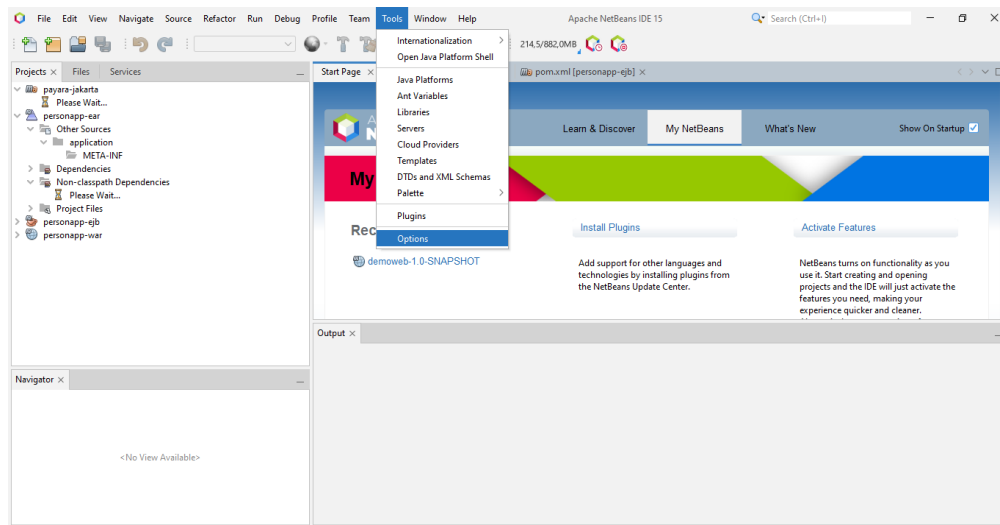
8. Damos en generate y guardamos el archivo producido

The screenshot shows a file download dialog box. The filename is "personapp-spring-boot" and the file type is "Archivo ZIP". There are two buttons at the bottom: "Guardar" (Save) and "Cancelar" (Cancel).

9. Luego, al guardar el archivo damos clic derecho y seleccionamos la opción extraer

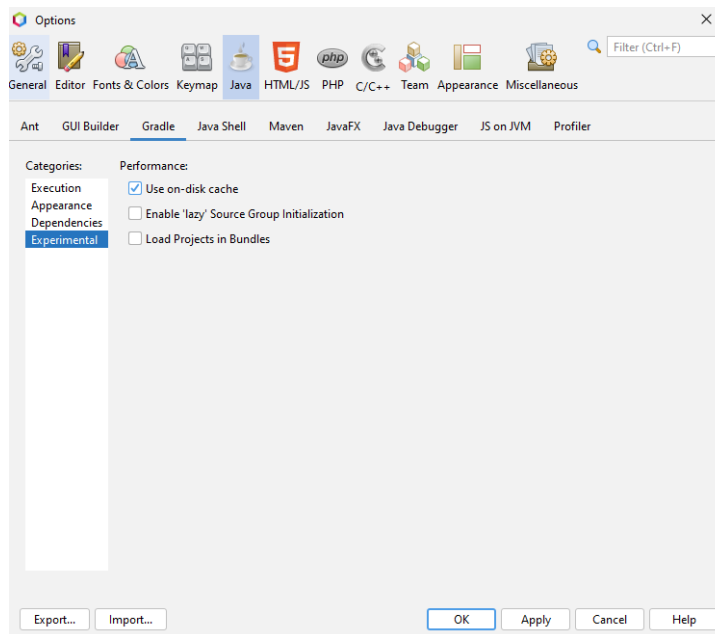
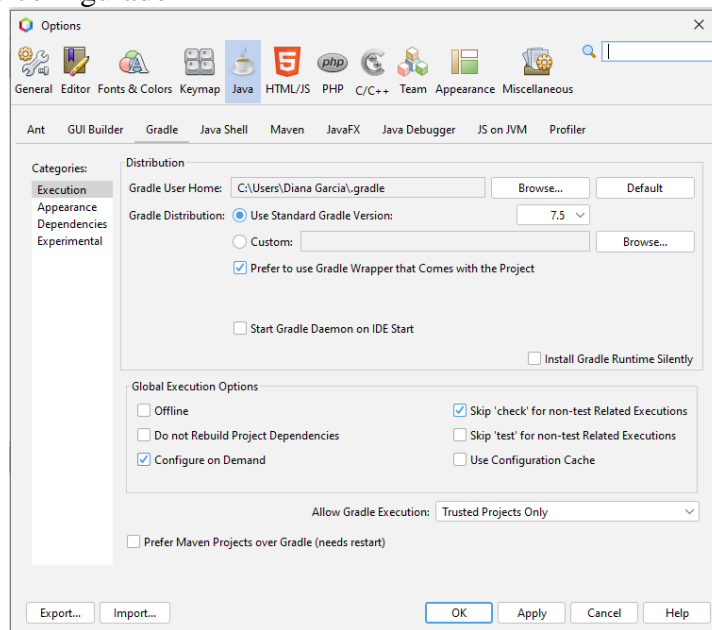


10. Abrimos Apache NetBeans IDE, seleccionamos la pestaña tools, en la opción options

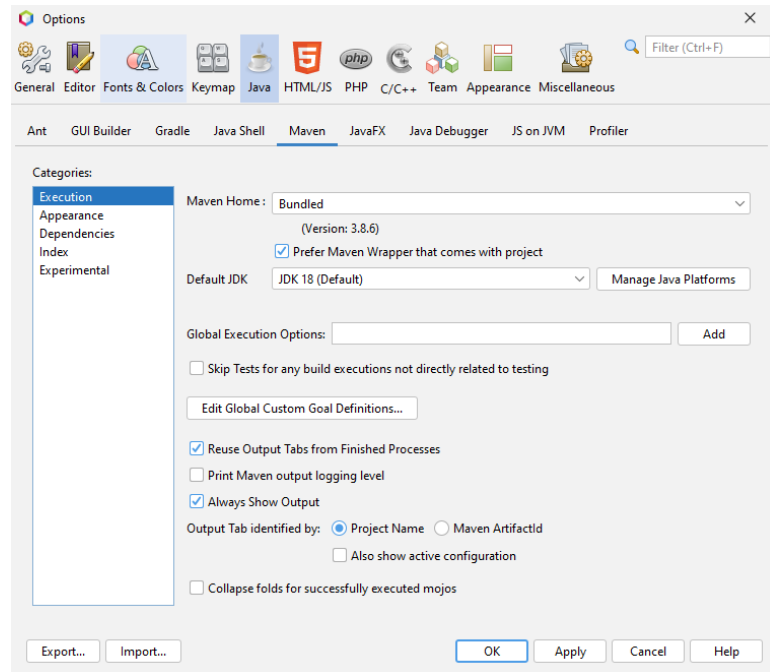




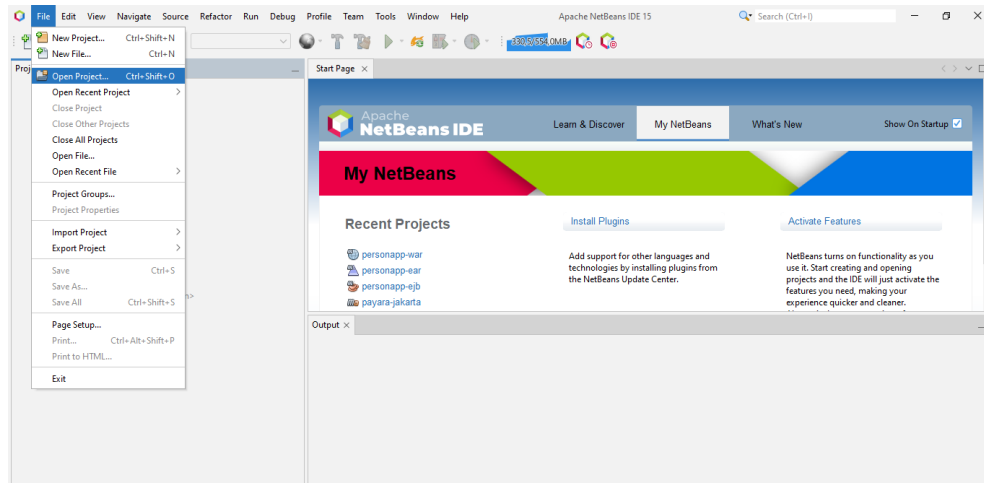
11. Nos vamos a la pestaña Java, opción gradle, para comprobar si todo está correctamente configurado

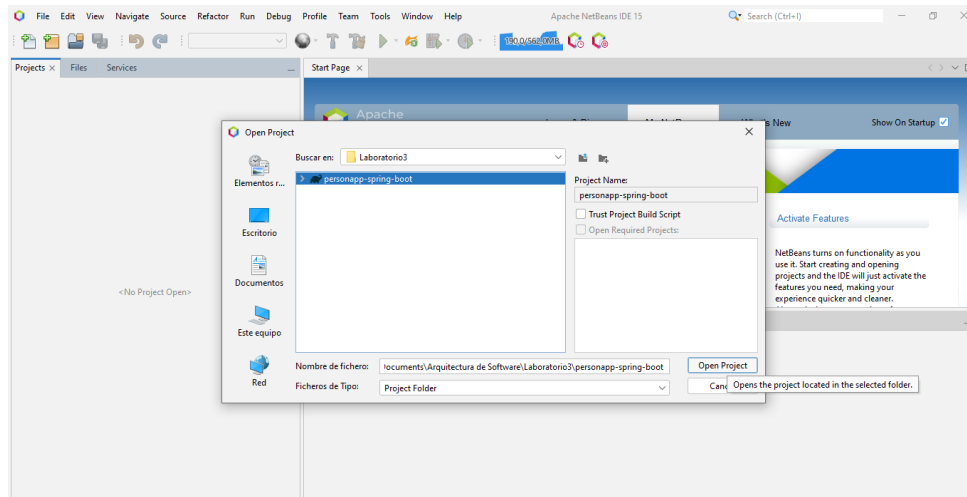


12. Comprobamos si está correcto el jdk

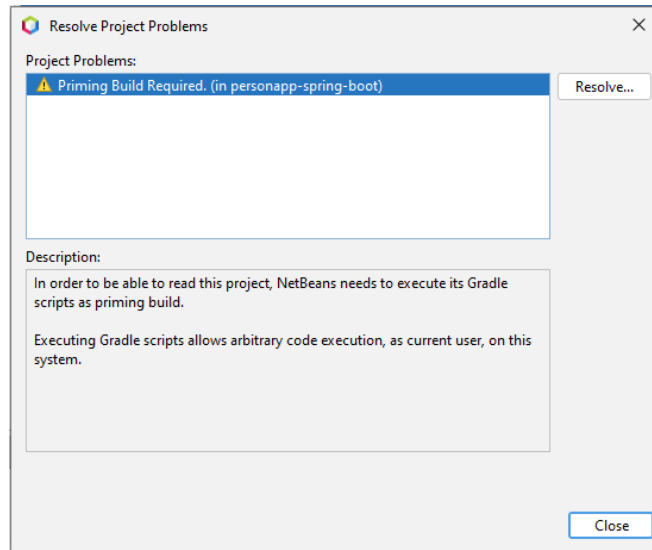


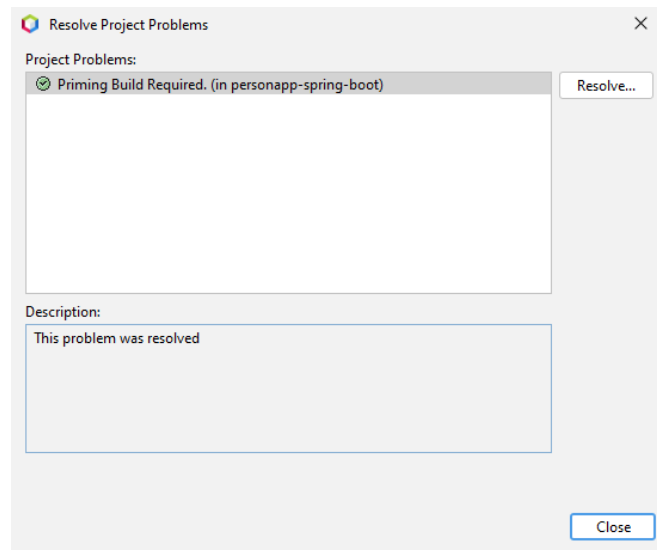
### 13. Abrimos el proyecto generado por spring en el Apache NetBeans



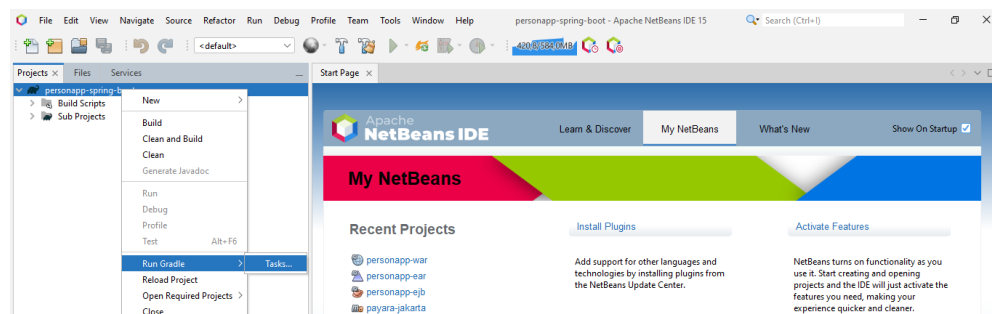


14. Aparecerá una pestaña emergente mostrando un problema, le damos clic en resolve y esperamos a que aparezca que el problema se solucionó

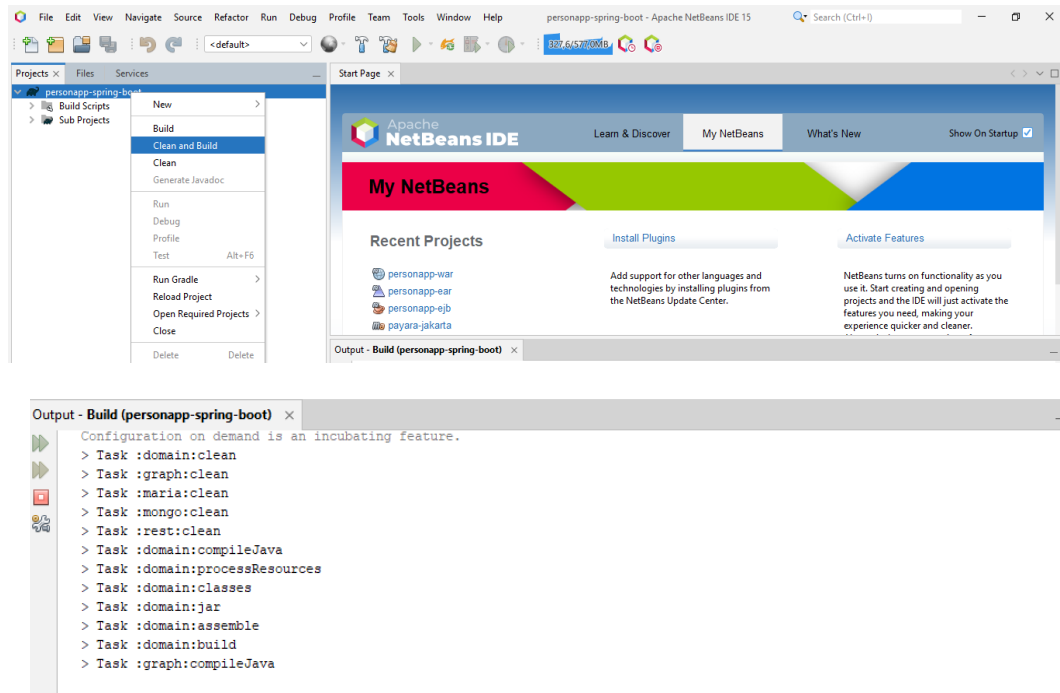




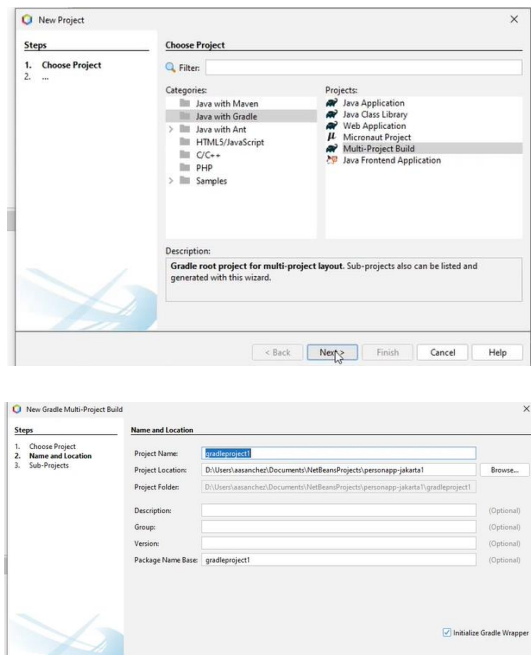
15. Damos clic derecho en el proyecto, en run gradle y en tasks

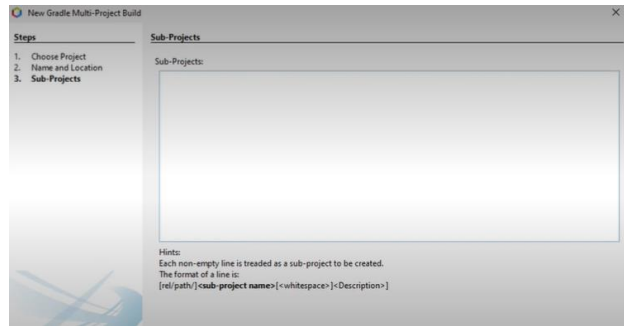


16. Damos clic derecho en el proyecto, en clean and build

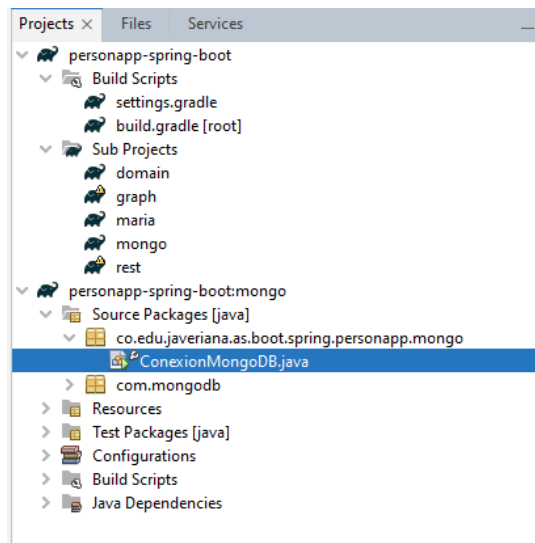


17. Creamos en Apache Netbeans un nuevo proyecto donde seleccionaremos Java with Gradle, Multi-Project Build y damos en siguiente, le dejamos el nombre por defecto y en siguiente





18. Luego, esperamos que se cree el proyecto
19. Realizamos ahora la configuración para unir la base de datos que en este caso será la conexión a MongoDB
20. Para esto, damos clic en la sección Mongo que aparece en el proyecto creado anteriormente y nos dirigimos a src/main/java en el archivo llamado ConexionMongoDB donde digitaremos el código para comprobar la conexión a la base de datos



21. Ejecutamos la base de datos y nos mostrará la que se ha creado

```
INFO: Opened connection [connectionId{localValue:2, serverValue:63}] to localhost:27017
0 - Personaapp-db
1 - Prueba_concepto
2 - admin
3 - basedatos2
4 - config
5 - local

BUILD SUCCESS
```

22. La base de datos se creó utilizando MongoDB Compass, como ya se había descargado anteriormente, sólo se abrió el programa

23. Al abrir el programa debemos elegir la conexión a la cual queremos que pertenezca la base de datos que vamos a utilizar, en este caso, se utilizó la conexión de manera local utilizando la dirección de local host predeterminada

**New Connection**  
Connect to a MongoDB deployment

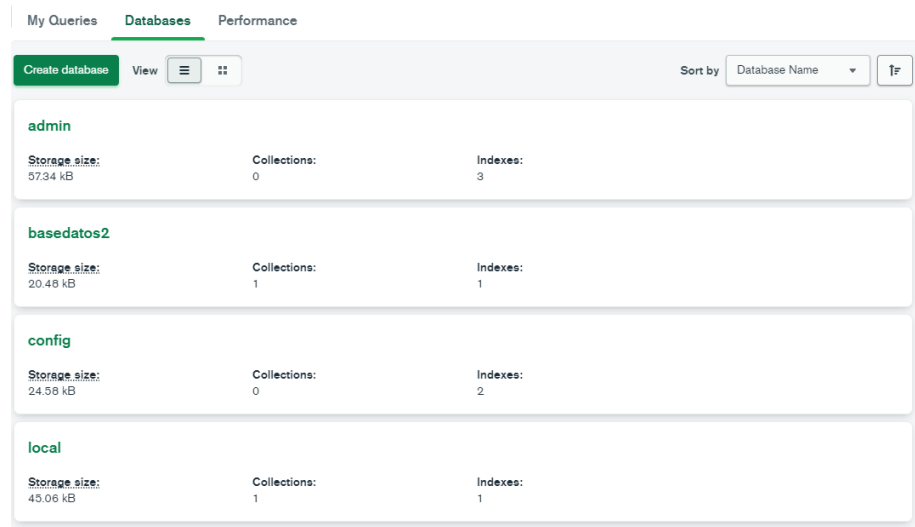
URI ⓘ Edit Connection String

mongodb://localhost:27017

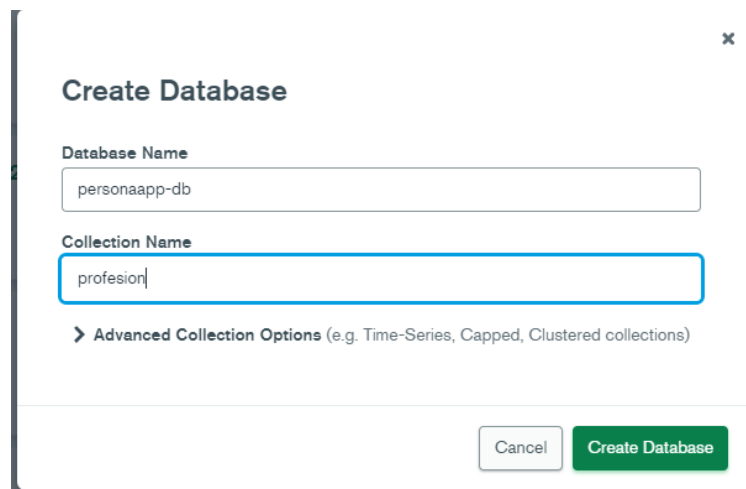
> Advanced Connection Options

Save Save & Connect Connect

24. Al darle Connect aparecerán todas las bases de datos que se han creado anteriormente, daremos clic a crear una nueva base de datos, ponemos el nombre que en este caso es personaapp-db y el nombre de la tabla que se inició con profesión tal y como lo dice el modelo dado en clase



My Queries Databases Performance		
Create database View [Menu Icon] [Grid Icon]		
Sort by Database Name [Dropdown] [Filter Icon]		
<b>admin</b>		
Storage size: 57.34 kB	Collections: 0	Indexes: 3
<b>basedatos2</b>		
Storage size: 20.48 kB	Collections: 1	Indexes: 1
<b>config</b>		
Storage size: 24.58 kB	Collections: 0	Indexes: 2
<b>local</b>		
Storage size: 45.06 kB	Collections: 1	Indexes: 1



### Create Database

Database Name

personaapp-db

Collection Name

profesion

> Advanced Collection Options (e.g. Time-Series, Capped, Clustered collections)

Cancel Create Database

25. Al tener la colección creada se agregará la información dando clic en ADD DATA, en la opción de insertar un documento donde se escribirá en el formato correspondiente y, además, se crearán las nuevas tablas que conforman el modelo



Insert to Collection Personaapp-db.profesion

VIEW 

{}

1

2

3

4

5

6

7

8

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

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

1000

1001

1002

1003

1004

1005

1006

1007

1008

1009

1010

1011

1012

1013

1014

1015

1016

1017

1018

1019

1020

1021

1022

1023

1024

1025

1026

1027

1028

1029

1030

1031

1032

1033

1034

1035

1036

1037

1038

1039

1040

1041

1042

1043

1044

1045

1046

1047

1048

1049

1050

1051

1052

1053

1054

1055

1056

1057

1058

1059

1060

1061

1062

1063

1064

1065

1066

1067

1068

1069

1070

1071

1072

1073

1074

1075

1076

1077

1078

1079

1080

1081

1082

1083

1084

1085

1086

1087

1088

1089

1090

1091

1092

1093

1094

1095

1096

1097

1098

1099

1100

1101

1102

1103

1104

1105

1106

1107

1108

1109

1110

1111

1112

1113

1114

1115

1116

1117

1118

1119

1120

1121

1122

1123

1124

1125

1126

1127

1128

1129

1130

1131

1132

1133

1134

1135

1136

1137

1138

1139

1140

1141

1142

1143

1144

1145

1146

1147

1148

1149

1150

1151

1152

1153

1154

1155

1156

1157

1158

1159

1160

1161

1162

1163

1164

1165

1166

1167

1168

1169

1170

1171

1172

1173

1174

1175

1176

1177

1178

1179

1180

1181

1182

1183

1184

1185

1186

1187

1188

1189

1190

1191

1192

1193

1194

1195

1196

1197

1198

1199

1200

1201

1202

1203

1204

1205

1206

1207

1208

1209

1210

1211

1212

1213

1214

1215

1216

1217

1218

1219

1220

1221

1222

1223

1224

1225

1226

1227

1228

1229

1230

1231

1232

1233

1234

1235

1236

1237

1238

1239

1240

1241

1242

1243

1244

1245

1246

1247

1248

1249

1250

1251

1252

1253

1254

1255

1256

1257

1258

1259

1260

1261

1262

1263

1264

1265

1266

1267

1268

1269

1270

1271

1272

1273

1274

1275

1276

1277

1278

1279

1280

1281

1282

1283

1284

1285

1286

1287

1288

1289

1290

1291

1292

1293

1294

1295

1296

1297

1298

1299

1300

1301

1302

1303

1304

1305

1306

1307

1308

1309

1310

1311

1312

1313

1314

1315

1316

1317

1318

1319

1320

1321

1322

1323

1324

1325

1326

1327

1328

1329

1330

1331

1332

1333

1334

1335

1336

1337

1338

1339

1340

1341

1342

1343

1344

1345

1346

1347

1348

1349

1350

1351

1352

1353

1354

1355

1356

1357

1358

1359

1360

1361

1362

1363

1364

1365

1366

1367

1368

1369

1370

1371

1372

1373

1374

1375

1376

1377

1378

1379

1380

1381

1382

1383

1384

1385

1386

1387

1388

1389

1390

1391

1392

1393

1394

1395

1396

1397

1398

1399

1400

1401

1402

1403

1404

1405

1406

1407

1408

1409

1410

1411

1412

1413

1414

1415

1416

1417

1418

1419

1420

1421

1422

1423

1424

1425

1426

1427

1428

1429

1430

1431

1432

1433

1434

1435

1436

1437

1438

1439

1440

1441

1442

1443

1444

1445

1446

1447

1448

1449

1450

1451

1452

1453

1454

1455

1456

1457

1458

1459

1460

1461

1462

1463

1464

1465

1466

1467

1468

1469

1470

1471

1472

1473

1474

1475

1476

1477

1478

1479

1480

1481

1482

1483

1484

1485

1486

1487

1488

1489

1490

1491

1492

149

### Personaapp-db.estudios

Documents Aggregations Schema Explain Plan Indexes Validation

**FILTER** { field: 'value' }

ADD DATA VIEW

Displaying

```
{
  "_id": ObjectId("637134e2016c5644638e81ec"),
  "id_prof": 1723244,
  "cc_per": "José",
  "fecha": "2000-01-01T05:00:00.000+00:00",
  "univer": "hhhsjkcxcjxc"
}
```

### Personaapp-db.persona

Documents Aggregations Schema Explain Plan Indexes Validation

**FILTER** { field: 'value' }

ADD DATA VIEW

```
{
  "_id": ObjectId("6371357d016c5644638e81f0"),
  "cc": 1723244,
  "nombre": "José",
  "apellido": "sanasndjfdijfidf",
  "genero": "M,F",
  "edad": 1233
}
```

### Personaapp-db.telefono

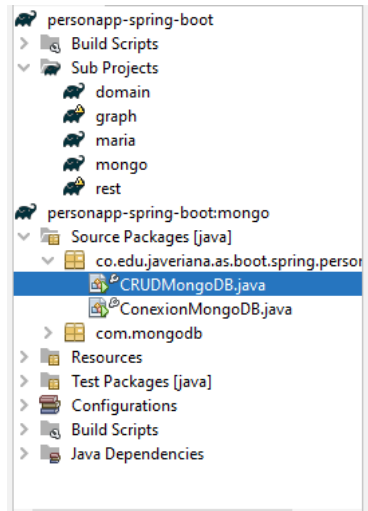
Documents Aggregations Schema Explain Plan Indexes Validation

**FILTER** { field: 'value' }

ADD DATA VIEW

```
{
  "_id": ObjectId("637135e5016c5644638e81f4"),
  "num": 1723244,
  "oper": "Claro",
  "duenio": 1234445
}
```

26. Al tener todas las tablas necesarias se realizará la conexión directa a la base de datos en el proyecto de NetBeans
27. Para generar la conexión volveremos al módulo src/main/java y crearemos un nuevo archivo java llamado CRUDMongoDB donde mostraremos la información de las tablas



```

public class CRUDMongoDB {
    public static void main(String[] args) {

        MongoClient mongo = crearConexion();

        // SI NO EXISTE LA BASE DE DATOS LA CREAMOS
        if(mongo != null) {
            DB db = mongo.getDB( "pruebas:Personaapp-db");

            mostrarColeccion(db, persona: "persona");
        }
    }

    public static MongoClient crearConexion() {
        System.out.println( "PRUEBA CONEXION MONGODB");

        MongoClient mongo = null;

        mongo = new MongoClient( localhost:"localhost", i:27017);

        return mongo;
    }

    public static void mostrarColeccion(DB db, String coleccion) {
        DBCollection colec = db.getCollection(coleccion);

        DBCursor cursor = colec.find();

        while(cursor.hasNext()) {
            System.out.println( " " + cursor.next().get("nombre") + " - " + cursor.curr().get("pais"));
        }
    }
}

```

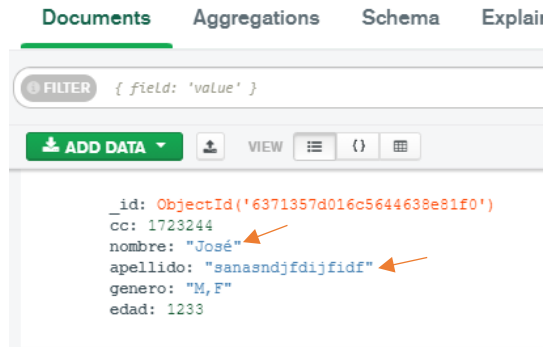
```

PRUEBA CONEXION MONGODB
nov. 13, 2022 3:48:53 P. M. com.mongodb.diagnostics.logging.JULLogger log
INFO: Cluster created with settings (hosts=[localhost:27017], mode=SHINGLE, requiredClusterType=UNKNOWN, serverSelectionTimeout='30000 ms', maxWaito
nov. 13, 2022 3:48:53 P. M. com.mongodb.diagnostics.logging.JULLogger log
INFO: Cluster description not yet available. Waiting for 30000 ms before timing out
nov. 13, 2022 3:48:53 P. M. com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:1, serverValue:64}] to localhost:27017
nov. 13, 2022 3:48:53 P. M. com.mongodb.diagnostics.logging.JULLogger log
INFO: Monitor thread successfully connected to server with description ServerDescription(address=localhost:27017, type=STANDALONE, state=CONNECTED,
nov. 13, 2022 3:48:53 P. M. com.mongodb.diagnostics.logging.JULLogger log
INFO: Opened connection [connectionId{localValue:2, serverValue:65}] to localhost:27017
* José - sanandjdfjdfdf

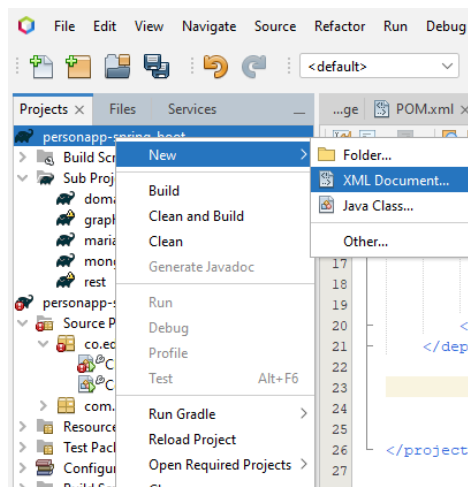
```

BUILD SUCCESS

## Personaapp-db.persona



28. Ya teniendo la conexión directa con Mongo, crearemos un archivo llamado pom.xml donde se guardarán todas las dependencias que se utilizarán en el desarrollo del proyecto. Para esto, se dará clic derecho en el nombre del proyecto, opción new file, de tipo xml



```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>co.edu.javeriana.as.boot.spring.personapp</groupId>
    <artifactId>ConexionMongoDB</artifactId>
    <version>1.0-SNAPSHOT</version>
    <packaging>jar</packaging>
    <properties>
        <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
        <maven.compiler.source>1.8</maven.compiler.source>
        <maven.compiler.target>1.8</maven.compiler.target>
    </properties>
    <dependencies>
        <!-- https://mvnrepository.com/artifact/org.mongodb/mongo-java-driver -->
        <dependency>
            <groupId>org.mongodb</groupId>
            <artifactId>mongo-java-driver</artifactId>
            <version>3.12.6</version>
        </dependency>
    </dependencies>
</project>
```

29. Crearemos ahora la conexión con la base de datos de MariaDB. Primero buscamos el archivo que se mostró al principio de este documento



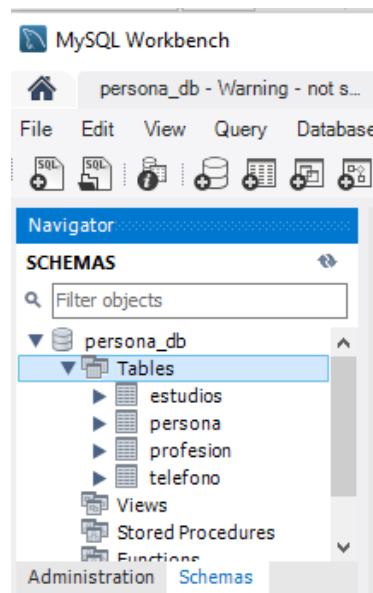
The screenshot shows the Maven Central page for the MariaDB Java Client 3.0.9. The page includes a breadcrumb trail: Home » org.mariadb.jdbc » mariadb-java-client » 3.0.9. The main title is "MariaDB Java Client » 3.0.9" with the subtitle "JDBC driver for MariaDB and MySQL". Below this is a table with the following information:

License	LGPL 2.1
Categories	JDBC Drivers
Tags	database   sql   jdbc   driver   mariadb   client   mysql
Organization	mariadb.com
HomePage	<a href="https://mariadb.com/kb/en/mariadb/about-mariadb-connector-j/">https://mariadb.com/kb/en/mariadb/about-mariadb-connector-j/</a>
Date	Nov 08, 2022
Files	<a href="#">pom (1.7 KB)</a>   <a href="#">jar (577 KB)</a>   <a href="#">View All</a>
Repositories	Central
Ranking	#614 in MvnRepository (See Top Artifacts) #7 in JDBC Drivers
Used By	668 artifacts

Below the table are tabs for various build systems: Maven, Gradle, Gradle (Short), Gradle (Kotlin), SBT, Ivy, Grape, Leiningen, and Buildr. At the bottom, there is a code block showing the Maven coordinates:

```
// https://mvnrepository.com/artifact/org.mariadb.jdbc/mariadb-java-client
implementation group: 'org.mariadb.jdbc', name: 'mariadb-java-client', version: '3.0.9'
```

30. Después, nos dirigimos a la aplicación MySQL Workbench para crear la base de datos, como fue realizada en el laboratorio número uno, no fue necesario realizar las tablas.



31. En Apache NetBeans seleccionamos la pestaña maria, en el componente de mapper generamos el código de conexión de MariaDB con la información de la base de datos creada en MySQL Workbench

```

: co.edu.javeriana.as.boot.spring.personapp.domain.model.Persona;
: co.edu.javeriana.as.boot.spring.personapp.maria.entity.PersonaEntity;

: java.sql.*;
: java.util.logging.Level;
: java.util.logging.Logger;

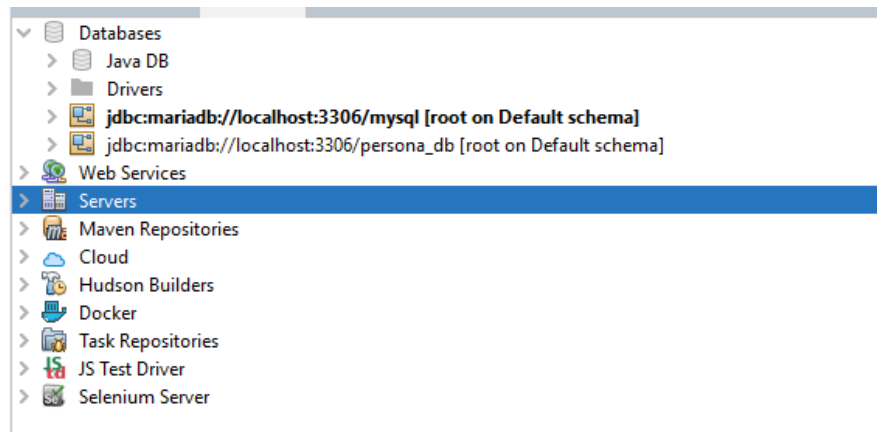
: class PersonaMapper {

: protected Persona toPersonaFromPersonaEntity(PersonaEntity pe){
:     return null;
: }

: public static void main(String[] args){

:     try {
:         Connection c = DriverManager.getConnection( url: "jdbc:mariadb://localhost:3306/persona_db", user: "root", password: "12345");
:         Statement s = (Statement) c.createStatement();
:         ResultSet res = s.executeQuery( select__from_profesion: "select * from profesion");
:         while (res.next()){
:             System.out.println( s: res.getString( sstring: "nom"));
:         }
:     } catch (SQLException e) {
:         System.out.println( s: "Error");
:         System.out.println( s: e);
:     }
: }

```



select * from profesion X				
Max. rows: 100   Fetched Rows: 3				
#	id	nom	des	
1	101	Franklin1	Garcia	
2	202	Frankli	Garci	
3	203	Frankl	Garc	

32. Y así, podemos observar toda la solución del laboratorio donde en cada sección se hacían los procesos correspondientes explicando, además, la manera en que se realizó paso a paso.

## Conclusiones y lecciones aprendidas

Se puede concluir que existen muchas maneras de enlazar los datos desde varios lugares que hacen más fácil el proceso. Cada una de las herramientas tiene sus propias características y en este caso, la ayuda de Mongo Compass para realizar la conexión de mongo en el proyecto a la base de datos y para hacer cada una de las funciones fue de fácil implementación.

Lo que más nos gustó del laboratorio es la forma tan diversa en que uno puede desarrollar algún proyecto teniendo varias alternativas especificando sus características al momento de realizar el proceso de la programación de las funciones. También, la forma en que en una sola herramienta se puedan implementar miles de aplicaciones dentro de un mismo espacio nos muestra que podemos facilitarnos mucho la vida en el desarrollo con miles de características si se necesita.

Esto deja una reflexión de que hay que abarcar más estudios, hay que aprender a utilizar cada una de las herramientas para tener la opción de escoger por facilidad e implementación, esto permite un mayor conocimiento, aprender nuevas cosas cada día, salir de la zona de confort y tener la motivación de seguir mejorando.

## Referencias

- [1]. ¿Cómo se utiliza Github pages? - Aprende sobre desarrollo web | MDN. Developer.mozilla.org. (2022). Retrieved 15 October 2022, from [https://developer.mozilla.org/es/docs/Learn/Common\\_questions/Using\\_Github\\_pages](https://developer.mozilla.org/es/docs/Learn/Common_questions/Using_Github_pages).
- [2]. Comenzar con GitHub Desktop. GitHub Docs. (2022). Retrieved 15 October 2022, from <https://docs.github.com/es/desktop/installing-and-configuring-github-desktop/overview/getting-started-with-github-desktop>.
- [3]. MySQL Workbench download now ". MySQL. (n.d.). Retrieved October 28, 2022, from <https://www.mysql.com/products/workbench/>
- [4]. Becker, A. (n.d.). What's this? HeidiSQL. Retrieved October 28, 2022, from <https://www.heidisql.com/>
- [5]. NetBeans, A. (n.d.). Apache NetBeans 15. Welcome to Apache NetBeans. Retrieved October 28, 2022, from <https://netbeans.apache.org/>
- [6]. Download mariadb server. MariaDB.org. (2021, September 29). Retrieved October 28, 2022, from <https://mariadb.org/download/?t=repo-config>
- [7]. Rubenfa. (2014, February 3). *MongoDB: Qué es, cómo funciona y cuándo podemos usarlo (O no)*. MongoDB. Qué es, cómo funciona y cuándo podemos usarlo (o no). Retrieved November 13, 2022, from <https://www.genbeta.com/desarrollo/mongodb-que-es-como-functiona-y-cuando-podemos-usarlo-o-no>
- [8] What is MongoDB Compass? - MongoDB Compass. (n.d.). Retrieved November 13, 2022, from <https://www.mongodb.com/docs/compass/current/>