

Ubiquitous Systems and Environmental Intelligence

Matilde Cabrera González

Índice:

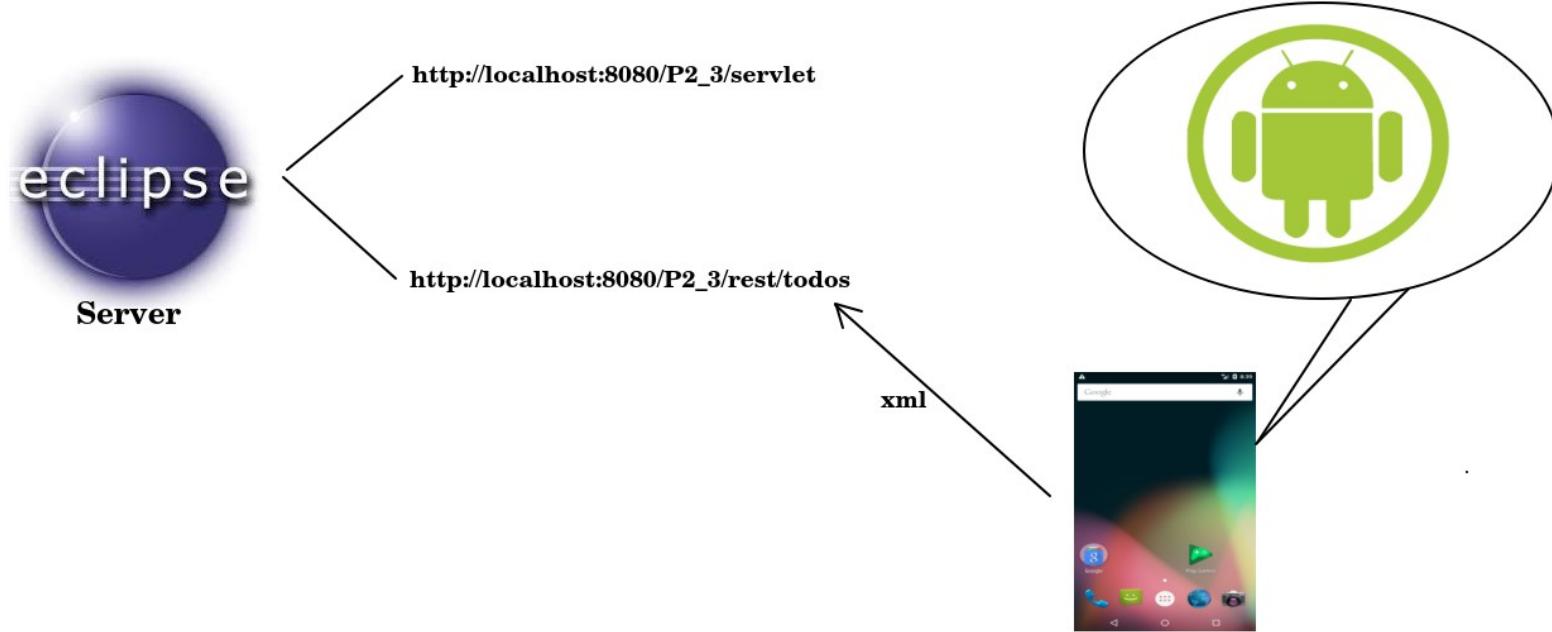
Introducción.

Android Studio

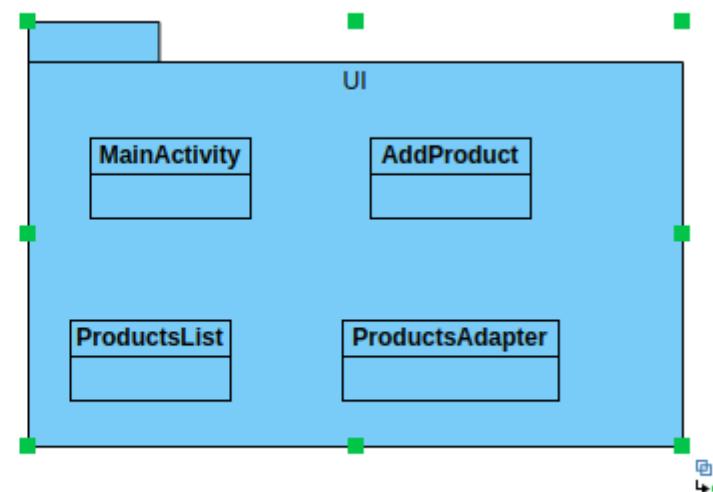
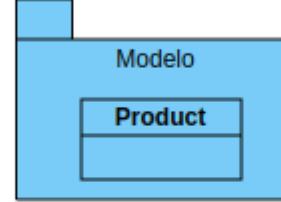
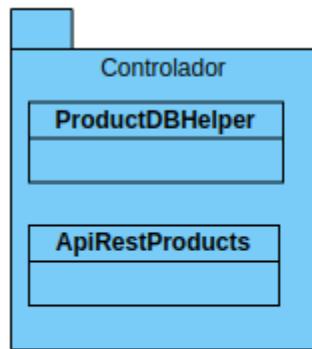
- Diagrama clases
- Pantallas
- Código

Server Rest Full Eclipse

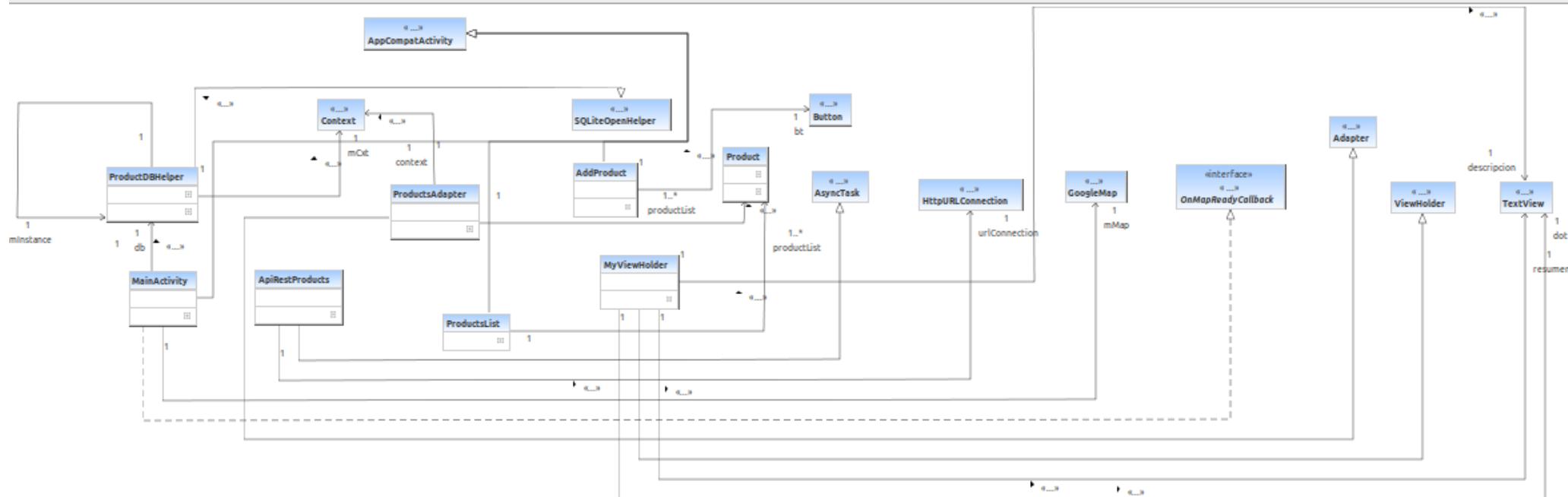
Introducción:

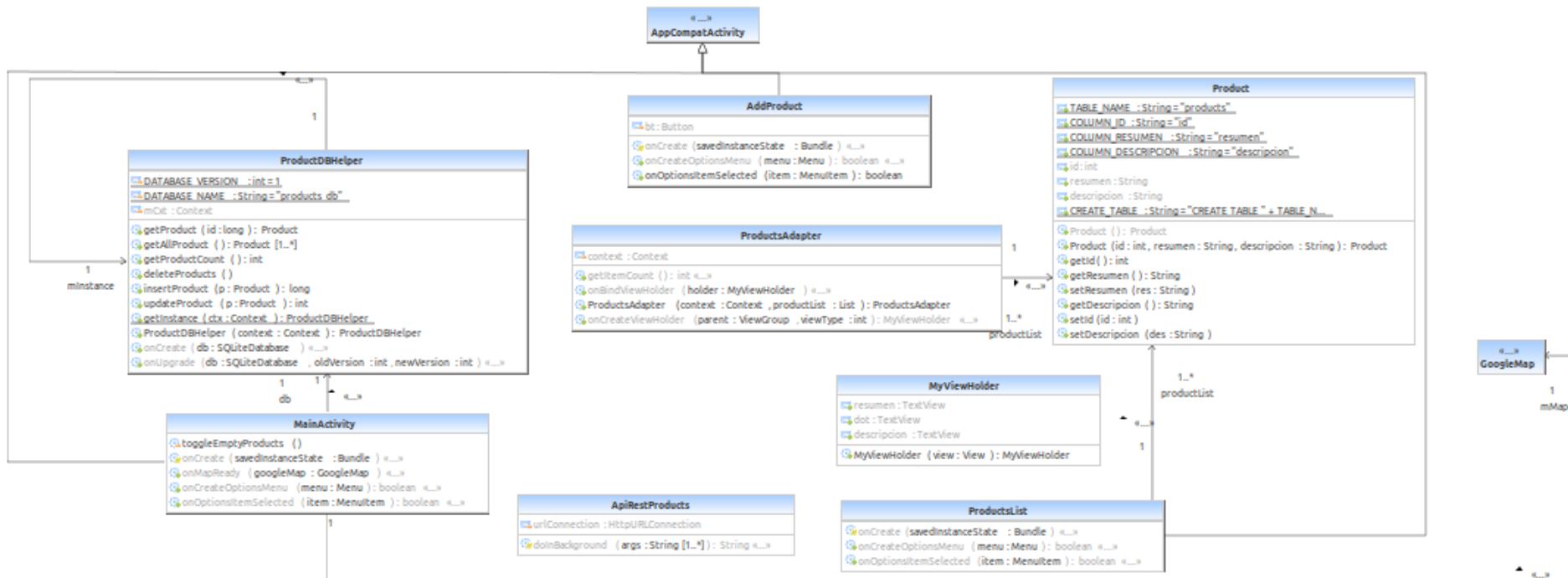


Android Studio

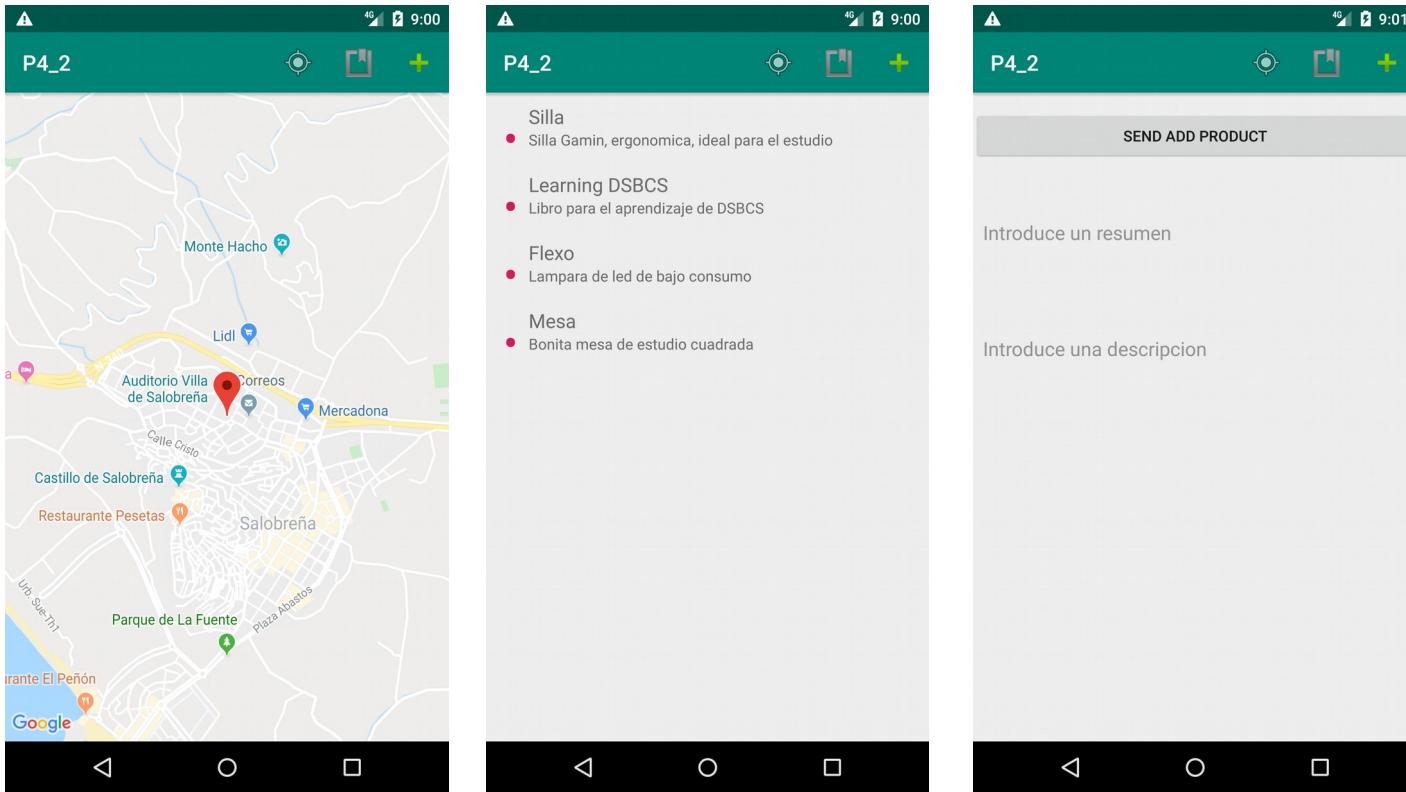


Android Studio





Android Studio



```

public class MainActivity extends AppCompatActivity implements OnMapReadyCallback{

    private GoogleMap mMap;
    public static ProductDBHelper db;

    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        db = ProductDBHelper.getInstance(this);
        toggleEmptyProducts();
        SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager()
            .findFragmentById(R.id.mapView);
        mapFragment.getMapAsync(this);
    }

    @Override
    public void onMapReady(GoogleMap googleMap) {
        mMap = googleMap;
        mMap.setMinZoomPreference(12);

        LatLng salo = new LatLng(36.7463, -3.58711);
        mMap.addMarker(new MarkerOptions().position(salo).title("Store in Salobreña"));
        mMap.moveCamera(CameraUpdateFactory.newLatLng(salo));
        CameraPosition cameraPosition = new
        CameraPosition.Builder().target(salo).zoom(15).build();
        googleMap.animateCamera(CameraUpdateFactory.newCameraPosition(cameraPosition));
    }

    private void toggleEmptyProducts() {

        if (db.getProductCount() <= 0) {
            try {
                new ApiRestProducts().execute().get();
            } catch (InterruptedException | ExecutionException e) {
                e.printStackTrace();
            }
        }
    }
}

```

```

// Menu Toolbar
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu items for use in the action bar
    getMenuInflater().inflate(R.menu.main_activity_action, menu);
    return super.onCreateOptionsMenu(menu);
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle presses on the action bar items
    switch (item.getItemId()) {
        case R.id.nav_products:
            Intent e = new Intent(this, ProductsList.class );
            startActivity(e);
            return true;
        case R.id.add_product:
            Intent a = new Intent(this, AddProduct.class );
            startActivity(a);
            return true;
        default:
            return super.onOptionsItemSelected(item);
    }
}

// Manifest

<uses-permission
    android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.INTERNET"/>

<meta-data
    android:name="com.google.android.geo.API_KEY"
    android:value="@string/google_maps_key" />

```

```

public class ProductsList extends AppCompatActivity {
    private List<Product> productList = new ArrayList<>();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_products_list);
        RecyclerView recyclerView = findViewById(R.id.recycler_view);
        productList.addAll(MainActivity.db.getAllProduct());
        ProductsAdapter mAdapter = new ProductsAdapter(this,
productList);
        RecyclerView.LayoutManager mLayoutManager = new
LinearLayoutManager(getApplicationContext());
        recyclerView.setLayoutManager(mLayoutManager);
        recyclerView.setAdapter(mAdapter);
    }
}

```

```

public class AddProduct extends AppCompatActivity {
    private Button bt;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_add_product);

        bt = findViewById(R.id.add_button);
        final Intent e = new Intent(this, ProductsList.class);

        // Add a OnClickListener object to bt.
        bt.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                EditText resField = findViewById(R.id.EditTextResumen);
                EditText desField=findViewById(R.id.EditTextDescripcion);
                if (!TextUtils.isEmpty(resField.getText().toString())){
                    Product p = new Product();
                    p.setResumen(resField.getText().toString());
                    p.setDescripcion(desField.getText().toString());
                    MainActivity.db.insertProduct(p);
                    startActivity(e);
                }
            }
        });
    }
}

```

```

public class ApiRestProducts extends AsyncTask<String, String, String> {

    private HttpURLConnection urlConnection;

    @Override
    protected String doInBackground(String... args) {
        StringBuilder result = new StringBuilder();
        try {
            URL url = new URL("http://10.0.2.2:8080/P2_3/rest/todos");
            urlConnection = (HttpURLConnection) url.openConnection();
            InputStream in = new BufferedInputStream(urlConnection.getInputStream());

            DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
            DocumentBuilder builder = factory.newDocumentBuilder();
            Document doc = builder.parse(in);
            Element raiz = doc.getDocumentElement();

            // Obtenemos todos los elementos
            NodeList items = raiz.getElementsByTagName("todo");
            for( int i = 0; i < items.getLength(); i++ ) {
                Node nodoproduc = items.item(i);
                Product product = new Product();

                // Recorremos todos los hijos que tenga
                for(int j = 0; j < nodoproduc.getChildNodes().getLength(); j++ ) {
                    Node nodoActual = nodoproduc.getChildNodes().item(j);

                    // Compruebo si es un elemento
                    if( nodoActual.getNodeType() == Node.ELEMENT_NODE ) {
                        if( nodoActual.getNodeName().equalsIgnoreCase("resumen") )
                            product.setResumen(nodoActual.getChildNodes().item(0)
                                .getNodeValue());
                        else if( nodoActual.getNodeName()
                            .equalsIgnoreCase("descripcion") )
                            product.setDescripcion(nodoActual.getChildNodes().item(0)
                                .getNodeValue());
                    }
                    result.append(product);
                    MainActivity.db.insertProduct(product);
                }
            }
            catch( Exception e ) {
                e.printStackTrace();
            }
        finally {
            urlConnection.disconnect();
        }
        return result.toString();
    }
}

```

```

public class ProductDBHelper extends SQLiteOpenHelper {

    private static ProductDBHelper mInstance = null;
    private static final int DATABASE_VERSION = 1;
    private static final String DATABASE_NAME = "products_db";
    private Context mContext;

    public static ProductDBHelper getInstance(Context ctx) {
        if (mInstance == null) {
            mInstance = new ProductDBHelper(ctx.getApplicationContext());
        }
        return mInstance;
    }

    public ProductDBHelper(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
        this.mContext = context;
    }

    // Creating Tables
    @Override
    public void onCreate(SQLiteDatabase db) {
        // create table
        db.execSQL(Product.CREATE_TABLE);
    }

    // Upgrading database
    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        // Drop older table if existed
        db.execSQL("DROP TABLE IF EXISTS " + Product.TABLE_NAME);

        // Create tables again
        onCreate(db);
    }

    public long insertProduct(Product p) {
        // get writable database as we want to write data
        SQLiteDatabase db = this.getWritableDatabase();

        ContentValues values = new ContentValues();
        // `id` will be inserted automatically.
        // no need to add them
        values.put(Product.COLUMN_RESUMEN, p.getResumen());
        values.put(Product.COLUMN_DESCRIPCION, p.getDescripcion());

        // insert row
        long id = db.insert(Product.TABLE_NAME, null, values);

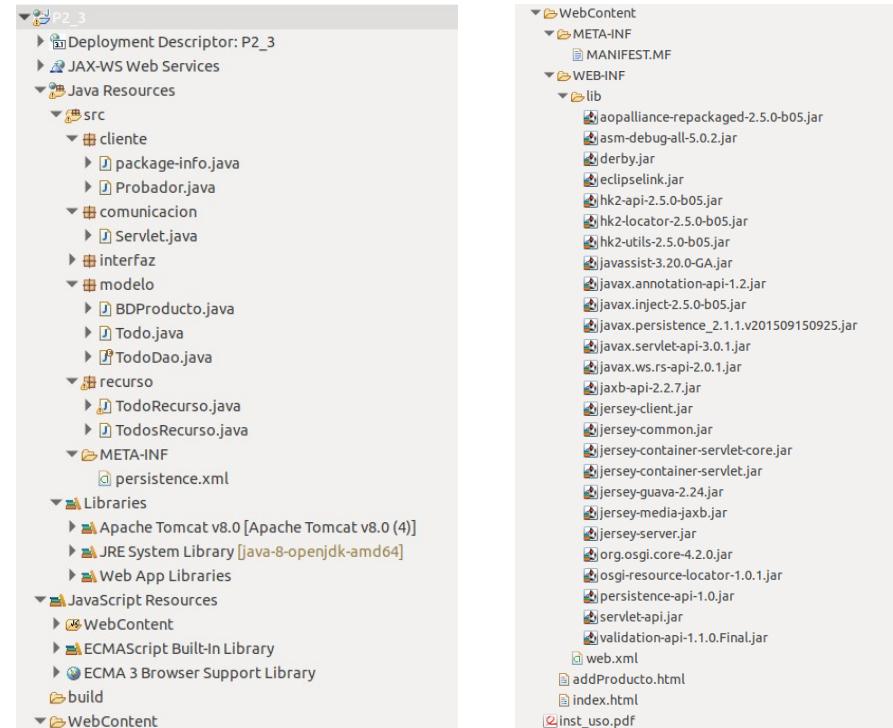
        // close db connection
        db.close();

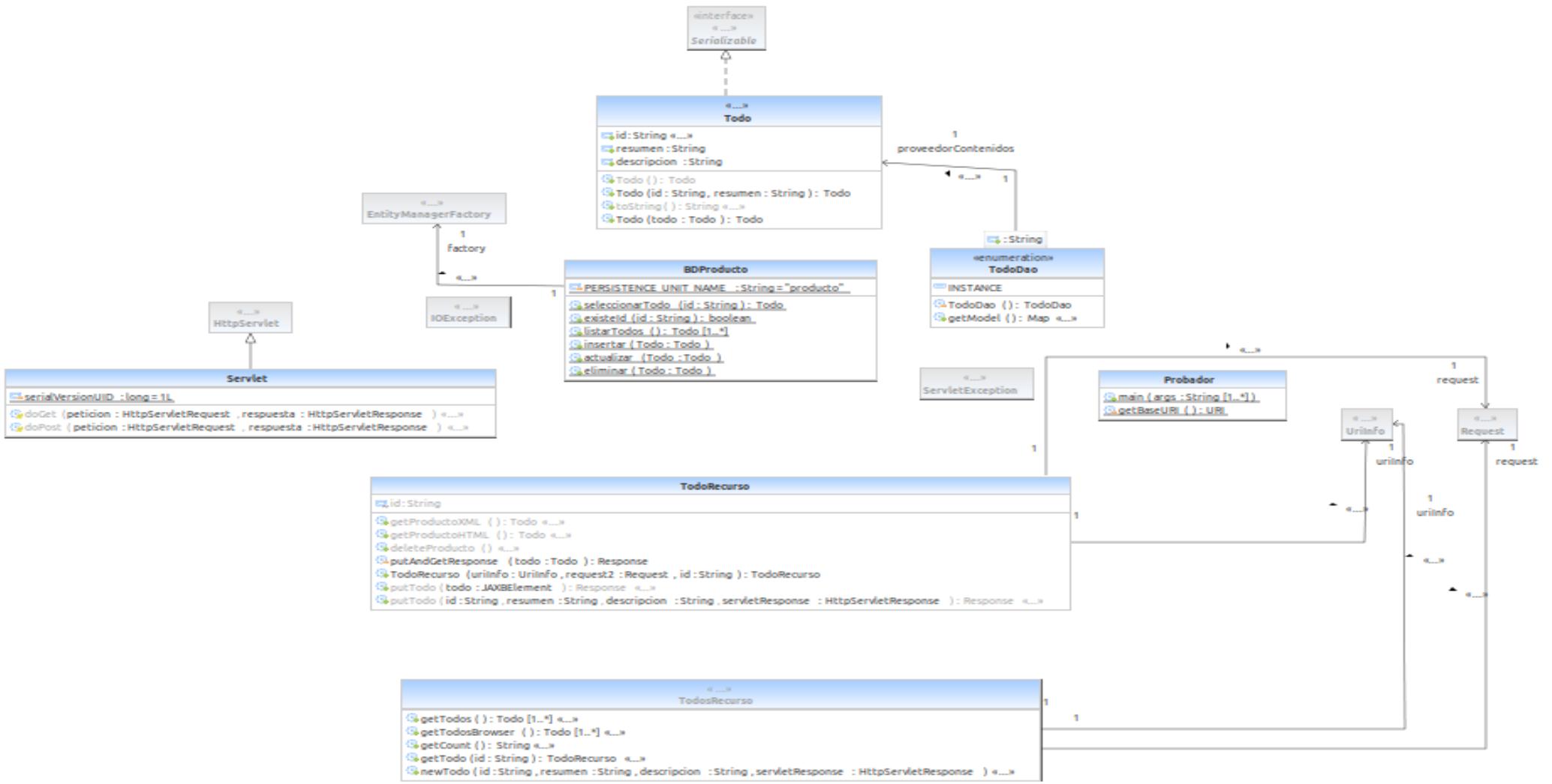
        // return newly inserted row id
        return id;
    }
}

```

Server Eclipse

- Base de datos relacionales con “Java Persistence API (JPA)”.
 - Definimos modelo.Todo como clase de conexión.
 - Base de datos creada en /tmp
- Despliegue de un SW con Jersey
 - Se definen dos mapping:
 - /rest/* → Manejo
 - /servlet → Descarga





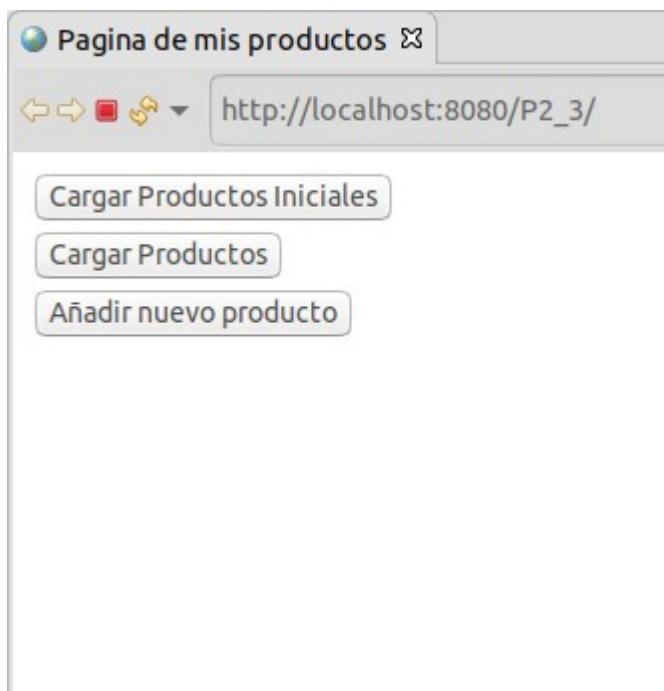
Probador de Server Eclipse

Run as Java Application

The screenshot shows the Eclipse IDE interface. On the left, the Project Explorer view displays the project structure for 'P2_3'. It includes a 'src' folder containing packages 'cliente', 'comunicacion', 'interfaz', 'modelo', 'recurso', and a file 'package-info.java'. It also contains a 'META-INF' folder with 'persistence.xml' and a 'Web App Libraries' section. The 'cliente' package is currently selected. On the right, the terminal window shows the output of running the application as a Java application. The output text is:

```
<terminated> Probador (2) [Java Application] /usr/lib/jvm/java-8-openjdk-amd64/bin/java (13 ene. 2020 12:50:47)
Codigo devuelto: 201
Mostrar como Texto XML Plano
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><todoes><todo><descripcion>Libro para el apr
Creacion de 1 formulario
Respuesta con el formulario 200
Contenidos del recurso, despues de enviar el elemento id=6
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><todoes><todo><descripcion>Libro para el apr
El elemento con id = 6 del recurso se ha eliminado
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><todoes><todo><descripcion>Libro para el apr
```

Web de Server Eclipse



The screenshot shows a web browser window titled "Pagina de mis productos". The address bar displays the URL "http://localhost:8080/P2_3/". Below the address bar are standard navigation buttons. On the left side of the page, there are two buttons: "Cargar Productos Iniciales" and "Cargar Productos". The main content area has a title "Productos" and a table displaying four products:

Id	Resumen	Descripcion
1	Learning DSBCS Libro para el aprendizaje de DSBCS	
2	Mesa	Bonita mesa de estudio cuadrada
3	Silla	Silla Gamin, ergonomica, ideal para el estudio
4	Flexo	Lampara de led de bajo consumo

At the bottom of the table is a button labeled "Añadir nuevo producto".

Web de Server Eclipse

Pagina de mis productos

http://localhost:8080/P2_3/

Cargar Productos Iniciales
Cargar Productos

Productos

Id	Resumen	Descripcion
1	Learning DSBCS	Libro para el aprendizaje de DSBCS
2	Mesa	Bonita mesa de estudio cuadrada
3	Silla	Silla Gamin, ergonomica, ideal para el estudio
4	Flexo	Lampara de led de bajo consumo

Añadir nuevo producto

ID: 5

resumen: Producto

descripcion: nuevo producto de prueba

Enviar

Pagina de mis productos

http://localhost:8080/P2_3/index.html

Cargar Productos Iniciales
Cargar Productos

Productos

Id	Resumen	Descripcion
1	Learning DSBCS	Libro para el aprendizaje de DSBCS
2	Mesa	Bonita mesa de estudio cuadrada
3	Silla	Silla Gamin, ergonomica, ideal para el estudio
4	Flexo	Lampara de led de bajo consumo
5	Producto	nuevo producto de prueba

Añadir nuevo producto

