



Sprint 4. Backend

“Consultorio Online”

Group 1 S33 - September 23th, 2021

Group members: Diego Lesmes, Miller Puentes, Nathalia Moreno, Nicolás Rojas & Pedro Ortiz

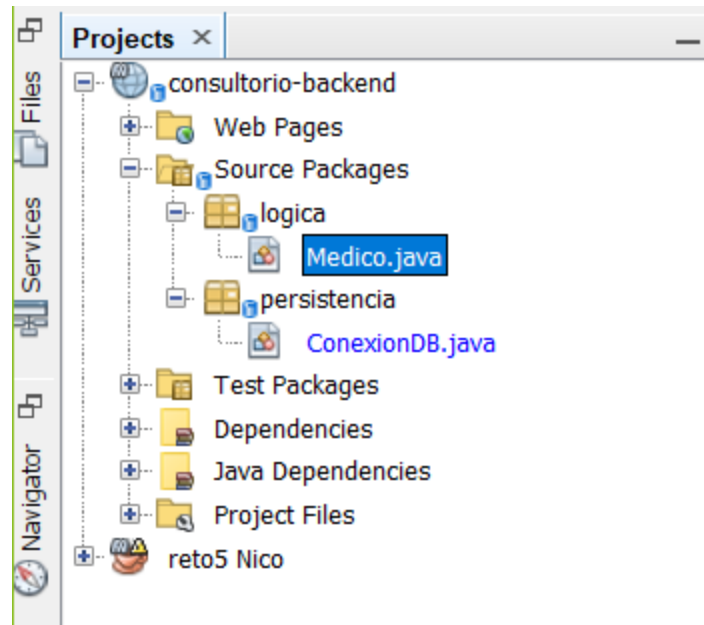
Delivery of the project source code for connection to the database and application logic

1. Design the structure of the application (Project settings in the development environment)

```
asf@asf-SVF15A17CLB:~/Documents/ConsultorioOnline_G1S33UIS$ tree
.
├── consultorio-backend
│   ├── nb-configuration.xml
│   ├── pom.xml
│   └── src
│       ├── main
│       │   ├── java
│       │   │   ├── logica
│       │   │   │   └── Medico.java
│       │   │   ├── persistencia
│       │   │   │   └── ConexionDB.java
│       │   └── webapp
│       │       ├── index.html
│       │       ├── META-INF
│       │       │   └── context.xml
│       │       └── WEB-INF
│       └── consultorio-frontend
│           ├── angular.json
│           ├── karma.conf.js
│           ├── package.json
│           ├── package-lock.json
│           ├── README.md
│           └── src
│               ├── app
│               │   ├── app.component.css
│               │   ├── app.component.html
│               │   ├── app.component.spec.ts
│               │   ├── app.component.ts
│               │   ├── app.module.ts
│               │   ├── app-routing.module.ts
│               │   └── componentes
│               │       ├── atendiendo-paciente
│               │       │   ├── atendiendo-paciente.component.html
│               │       │   └── atendiendo-paciente.component.ts
│               │       ├── componentes.module.ts
│               │       └── encabezados
│               └── ...
```



2. Design and code the classes for database connection and CRUD transactions for one (1) of the system modules according to the requirements defined in Sprint 1 and 2.



3. Connect the database and implement the necessary methods for the program's functionalities

[Class in Gitlab](#)



```
Start Page x ConexionBD.java x ConexionDB.java x Contacto.java x
Source History
1 ...5 lines
6 package persistencia;
7
8 /**
9  *
10  * @author nathalia.moreno
11  */
12 import java.sql.*;
13 public class ConexionDB {
14
15     private String url="";
16     private String DB_driver="";
17     private String host="";
18     private String db="";
19     private String username="";
20     private String password="";
21     public Connection con=null;
22     private Statement stmt=null;
23     private ResultSet rs=null;
24     private boolean local;
25
26     public ConexionDB () {
27         local=false;
28         DB_driver="com.mysql.cj.jdbc.Driver";
29         if(local) {
30             host="localhost:3306";
31             db="S33Grupol";
32             url="jdbc:mysql://" + host + "/" + db;
33             username="root";
34             password="root";
35         } else {
36             host="167.99.168.84:3306";
37             db="S33Grupol";
38             url="jdbc:mysql://" + host + "/" + db;
39             username="mision";
40             password="tics";
41         }
42     }
43 }
```



```
public ResultSet consultarBD(String sentencia) {  
    try{  
        stmt= con.createStatement();  
        rs= stmt.executeQuery(sentencia);  
    } catch (SQLException sqlex) {  
        System.out.println(sqlex.getMessage());  
    } catch (RuntimeException rex){  
        System.out.println(rex.getMessage());  
    } catch (Exception ex) {  
        System.out.println(ex.getMessage());  
    }  
    return rs;  
}
```

```
public boolean insertarBD(String sentencia){  
    try{  
        stmt = con.createStatement();  
        stmt.execute(sentencia);  
    } catch (SQLException | RuntimeException sqlex){  
        System.out.println("Error en insertar " + sqlex.getMessage());  
        return false;  
    }  
    return true;  
}
```

```
try {  
    Class.forName(DB_driver);  
} catch (ClassNotFoundException ex){  
    System.out.println("Error en el driver de la base de datos: " + ex);  
}  
  
try{  
    con= DriverManager.getConnection(url, username, password);  
    if(con!=null){  
        DatabaseMetaData meta= con.getMetaData();  
        System.out.println("Base de Datos Conectada " + meta.getDriverName());  
    }  
} catch (SQLException ex) {  
    System.out.println("Error en la conexión a la base de datos" + ex.getMessage());  
}
```

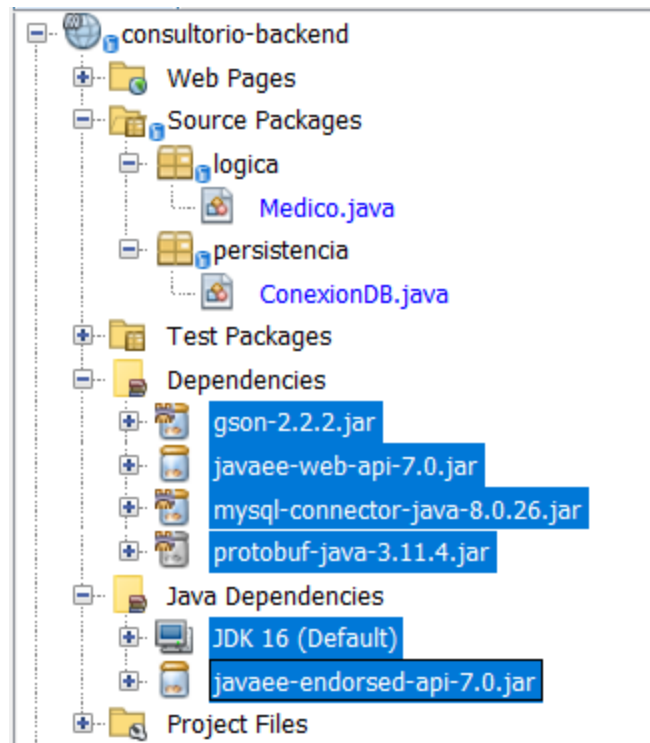


```
public boolean actualizarBD(String sentencia){
    try{
        stmt = con.createStatement();
        stmt.executeUpdate(sentencia);
    } catch(SQLException | RuntimeException sqllex){
        System.out.println("Error en actualizar " + sqllex.getMessage());
        return false;
    }
    return true;
}

public boolean borrarBD(String sentencia){
    try{
        stmt = con.createStatement();
        stmt.execute(sentencia);
    } catch(SQLException | RuntimeException sqllex){
        System.out.println("Error en borrar " + sqllex.getMessage());
        return false;
    }
    return true;
}

public static void main(String[] args){
    ConexionDB ejemplo = new ConexionDB();
}
}
```

4. The creation and configuration of a java web Application regarding backend components is evidenced.



5. The connection to the database is evidenced in the persistence layer.

Return to -> [3. Connect the database and implement the necessary methods for the program's functionalities](#)



6. At least one class is evidenced that represents a database entity in the application for the logical layer.

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package logica;

/**
 *
 * @author Nicolás Rojas
 */
import java.sql.ResultSet;
import java.util.ArrayList;
import java.sql.SQLException;
import persistencia.ConexionDB;

public class Medico {

    private int idMedico;
    private String primerNombre;
    private String segundoNombre;
    private String primerApellido;
    private String segundoApellido;
    private String usuario;
    private String contraseña;
    private String fotoUsuario;
    private String firma;
    private ConexionDB conexion;

    @Override
    public String toString() {
        return "Medico{" + "primerNombre=" + primerNombre + ", segundoNombre=" + segundoNombre + ", primerApellido=" + primerApellido + ", segundoApellido=" + segundoApellido + ", usuario=" + usuario + ", contrase\u00f1a=" + contraseña + ", fotoUsuario=" + fotoUsuario + ", firma=" + firma + '}';
    }

    public Medico() {
        this.conexion = new ConexionDB();
    }

    public Medico(String primerNombre, String segundoNombre, String primerApellido, String segundoApellido, String usuario, String contraseña, String fotoUsuario, String firma) {
        this.conexion = new ConexionDB();
        this.primerNombre = primerNombre;
        this.segundoNombre = segundoNombre;
        this.primerApellido = primerApellido;
        this.segundoApellido = segundoApellido;
        this.usuario = usuario;
        this.contrase\u00f1a = contraseña;
        this.fotoUsuario = fotoUsuario;
        this.firma = firma;
    }

    public int getIdMedico() {
        return idMedico;
    }

    public String getPrimerNombre() {
        return primerNombre;
    }

    public String getSegundoNombre() {
        return segundoNombre;
    }

    public String getPrimerApellido() {
        return primerApellido;
    }

    public String getSegundoApellido() {
        return segundoApellido;
    }

    public String getUsuario() {
        return usuario;
    }

    public String getContrase\u00f1a() {
        return contrase\u00f1a;
    }
}
```



```
}

public String getFotoUsuario() {
    return fotoUsuario;
}

public String getFirma() {
    return firma;
}

public void setIdMedico(int idMedico) {
    this.idMedico = idMedico;
}

public void setPrimerNombre(String primerNombre) {
    this.primerNombre = primerNombre;
}

public void setSegundoNombre(String segundoNombre) {
    this.segundoNombre = segundoNombre;
}

public void setPrimerApellido(String primerApellido) {
    this.primerApellido = primerApellido;
}

public void setSegundoApellido(String segundoApellido) {
    this.segundoApellido = segundoApellido;
}

public void setUsuario(String usuario) {
    this.usuario = usuario;
}

public void setContraseña(String contraseña) {
    this.contraseña = contraseña;
}
```

```
public void setFotoUsuario(String fotoUsuario) {
    this.fotoUsuario = fotoUsuario;
}

public void setFirma(String firma) {
    this.firma = firma;
}

public boolean guardarMedico() {
    String sentencia = "INSERT INTO medico (primerNombre, segundoNombre, primerApellido, segundoApellido, usuario, contraseña, fotoUsuario, firma) "
        + "VALUES ('" + this.primerNombre + "','" + this.segundoNombre + "','" + this.primerApellido + "','" + this.segundoApellido + "','"
        + this.usuario + "','" + this.contraseña + "','" + this.fotoUsuario + "','" + this.firma + "');"

    return this.conexion.insertarBD(sentencia);
}

public ArrayList<Medico> listaMedicos() {
    ArrayList<Medico> listaMedicos = new ArrayList<>();
    String sentencia = "SELECT * FROM medico;";
    ResultSet rs = this.conexion.consultarBD(sentencia);
    Medico medico = new Medico();
    try {
        while (rs.next()) {
            medico.setPrimerNombre(rs.getString("primerNombre"));
            medico.setSegundoNombre(rs.getString("segundoNombre"));
            medico.setPrimerApellido(rs.getString("primerApellido"));
            medico.setSegundoApellido(rs.getString("segundoApellido"));
            medico.setUsuario(rs.getString("usuario"));
            medico.setContraseña(rs.getString("contraseña"));
            medico.setFotoUsuario(rs.getString("fotoUsuario"));
            medico.setFirma(rs.getString("firma"));
            listaMedicos.add(medico);
        }
    } catch (SQLException ex) {
        System.out.println("Error en la consulta a la bd: " + ex.getMessage());
    }
    return listaMedicos;
}
```




```

public Medico obtenerMedico() {
    String sentencia = "SELECT * FROM medico WHERE usuario = '" + this.usuario + "'";
    ResultSet rs = this.conexion.consultarBD(sentencia);
    try {
        if (rs.next()) {
            this.primerNombre = rs.getString("primerNombre");
            this.segundoNombre = rs.getString("segundoNombre");
            this.primerApellido = rs.getString("primerApellido");
            this.segundoApellido = rs.getString("segundoApellido");
            this.usuario = rs.getString("usuario");
            this.contrasena = rs.getString("contrasena");
            this.fotoUsuario = rs.getString("fotoUsuario");
            this.firma = rs.getString("firma");
            return this;
        } else {
            return null;
        }
    } catch (SQLException ex) {
        System.out.println("Error en la consulta a la bd: " + ex.getMessage());
        return null;
    }
}

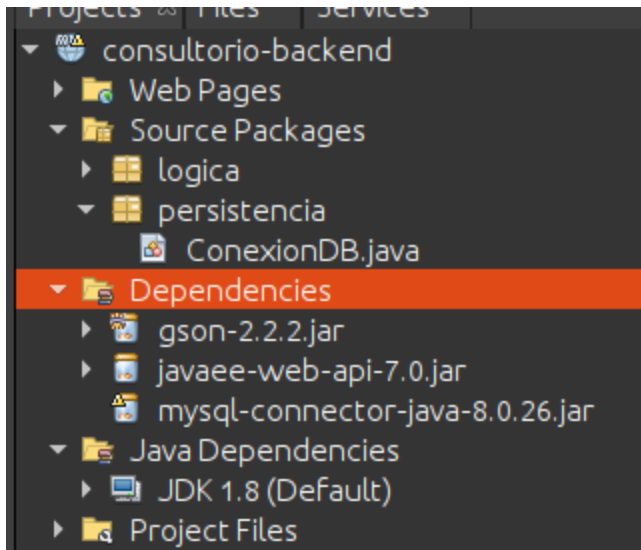
public boolean actualizarMedico(int id) {
    String sentencia = "UPDATE medico SET primerNombre = '" + this.primerNombre + "', segundoNombre = '" + this.segundoNombre
        + "', primerApellido = '" + this.primerApellido + "', segundoApellido = '" + this.segundoApellido
        + "', usuario = '" + this.usuario + "', contrasena = '" + this.contrasena + "', fotoUsuario = '" + this.fotoUsuario + "', firma = '" + this.firma + "'";
    return this.conexion.actualizarBD(sentencia);
}

public boolean borrarMedico(int idMedico) {
    String sentencia = "DELETE FROM medico WHERE idMedico = " + idMedico;
    return this.conexion.borrarBD(sentencia);
}

//
// public static void main(String[] args) {
//     Medico medico1 = new Medico("Miller", "", "Puentes", "", "MillenPuentes", "111111", "foto1111", "firma2222");
//     Medico medico2 = new Medico();
//     medico1.guardeMedico();
//     medico1.actualizarMedico(1);
//     medico2.borrarMedico(1);
//     System.out.println(medico2.listaMedicos());
//     System.out.println(medico1.obtenerMedico().toString());
// }

```

7. The configuration of project dependencies is evident.





8. The project is added to the social coding GitLab repository.

The screenshot shows the GitLab web interface for a repository named 'ConsultorioOnline_G1533UIS' by user 'Diego Lesmes'. The repository is located at 'gitlab.com/DLesmes/ConsultorioOnline_G1533UIS'. The interface includes a sidebar with navigation options like Project information, Repository, Issues, Merge requests, CI/CD, Security & Compliance, Deployments, Monitor, Infrastructure, Packages & Registries, Analytics, Wiki, Snippets, and Settings. The main content area displays repository statistics: 31 Commits, 6 Branches, 0 Tags, 512 KB Files, and 625 KB Storage. A security banner encourages using Static Application Security Testing (SAST). Below this, an 'Auto DevOps' section offers to automatically build, test, and deploy based on a predefined CI/CD configuration. The repository is currently on the 'main' branch. A recent commit by 'Clase médico agregada' (Diego Lesmes) is shown, with a commit hash of '0a355823'. At the bottom, there are buttons for 'Upload File', 'README', 'Add LICENSE', 'Add CHANGELOG', 'Add CONTRIBUTING', 'Add Kubernetes cluster', 'Set up CI/CD', 'Configure Integrations', and 'Add Security Testing'. A table at the bottom lists the commit details.





















Name	Last commit	Last update
consultorio-backend	Clase médico agregada	50 minutes ago

9. The teacher and tutor are added to the project created in the repository.

Members 7

Filter members

Account

Account	Source	Access granted	Access expires	Max role	Expiration
 Diego Lesmes It's you @DLesmes	Direct member	2 weeks ago by Diego Lesmes	No expiration set	Maintainer	Expiration date 
 Elkin Diaz @ediazMintic	Direct member	2 weeks ago by Diego Lesmes	No expiration set	Maintainer	Expiration date  
 Miller Puentes @millerpuentes	Direct member	2 weeks ago by Diego Lesmes	No expiration set	Maintainer	Expiration date  
 Nathalia Moreno Sánchez @nathy92moreno	Direct member	2 weeks ago by Diego Lesmes	No expiration set	Maintainer	Expiration date  
 Nicolás Rojas @Nico Rojas25	Direct member	2 weeks ago by Diego Lesmes	No expiration set	Maintainer	Expiration date  
 Pedro Ortiz @danJack010892	Direct member	2 weeks ago by Diego Lesmes	No expiration set	Maintainer	Expiration date  
 Tutor 11 @tutor11	Direct member	2 weeks ago by Diego Lesmes	No expiration set	Maintainer	Expiration date  



10. Participation in team meetings is shown, as well as communication through Jira channels and others supported on the Moodle platform.

The screenshot illustrates a collaborative development environment. The central focus is the Apache NetBeans IDE, where the `Medico.java` file is being edited. The code includes methods for setting and getting medical data, such as `setIdMedico`, `setPrimerNombre`, `setSegundoNombre`, `setPrimerApellido`, and `setSegundoApellido`. A 'Participants' window is open, showing the current session with five participants: Nathalia Moreno (Me), Nicolás Rojas, Diego Lesmes, Miller Puentes, and Pedro Ortiz. The bottom of the screen shows the Zoom interface, indicating that the session is active and participants can interact. The right sidebar displays a list of participants, with Nicolás Rojas highlighted, suggesting active communication and collaboration during the meeting.

Gitlab members commits



25 Sep, 2021 3 commits



Se modifico la clase medico agregando Operaciones CRUD.

Pedro Ortiz authored 1 minute ago



Code from class added

Diego Lesmes authored 41 minutes ago



Source test code

Diego Lesmes authored 42 minutes ago

24 Sep, 2021 1 commit



Se agregan operaciones CRUD a la clase médico.

Pedro Ortiz authored 23 hours ago

23 Sep, 2021 12 commits



Clase médico agregada

Diego Lesmes authored 1 day ago



Se coloca el nombre de la DB S33Grupo1

Nathalia Moreno Sánchez authored 1 day ago



Se añade conexión a la Base de Datos

Nathalia Moreno Sánchez authored 1 day ago



Se añade la conexión a la Base de Datos

Nathalia Moreno Sánchez authored 1 day ago



Se instalan las dependencias necesarias para el proyecto.

Miller Puentes authored 1 day ago



Merge branch 'main' of gitlab.com:DLesmes/ConsultorioOnline_G1S33UIS into main ...

Diego Lesmes authored 1 day ago



Creación del proyecto

Nicolás Rojas authored 1 day ago

Setting Spring 3



Jira Software Your work Projects Filters Dashboards People Apps Create

S33Grupo1 Software project

Roadmap Backlog Board Reports Issues Code Deployments Releases On-call Project pages Add shortcut Project settings

Projects / S33Grupo1

Sprint 4

Entrega del código fuente del proyecto para conexión a base de datos y lógica de la aplicación

Label Type

TO DO 4 ISSUES

TR_007_Tipo de aplicación
Desarrollo Varias
S33GROUP-18 1

TR_009_Lenguajes y frameworks
BackEnd FrontEnd
S33GROUP-19 13

HU_002_Como médico quisiera poder revisar mi agenda, programar una cita y modificar su estado según se necesite para cumplir con el cronograma de atención a pacientes.
BackEnd FrontEnd
Gestión_Proyectos Desarrollo Varias
S33GROUP-35 8

Crear Operaciones del CRUD para la tabla médico
S33GROUP-81

IN PROGRESS 4 ISSUES

HU_001_Como médico quisiera que al iniciar la aplicación, el sistema me pidiera un usuario y contraseña para acceder a la información.
BackEnd FrontEnd
Gestión_Proyectos Desarrollo Varias
S33GROUP-34 8

Actualizar Cronograma
S33GROUP-74

Enviar informe de seguimiento a Tutor
S33GROUP-75

Revisar las Tablas Creadas y modificar el campo
S33GROUP-76

PEER REVIEW 1 ISSUE

Crear la clase médico con sus respectivos métodos
S33GROUP-80

DONE 4 ISSUES

Crear Documento del Sprint 4
S33GROUP-73

Crear el proyecto en el repositorio (Backend)
S33GROUP-77

Crear la conexión a la BD
S33GROUP-79

Crear las dependencias del proyecto
S33GROUP-78

You're in a team-managed project [Learn more](#)

Burnup Report

Projects / S33Grupo1 / Reports

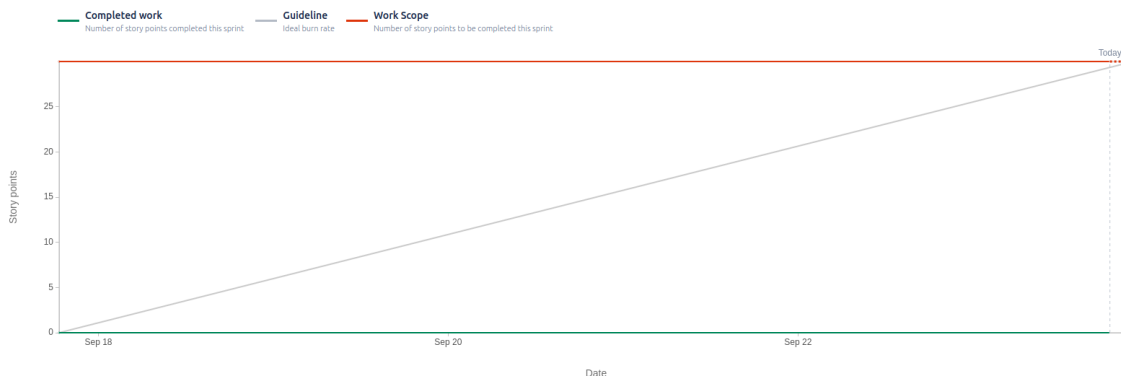
Burnup report

[How to read this](#)

Sprint Estimation field
Sprint 4 Story points

Date - 17 September 2021 - 23 September 2021

Sprint goal - Entrega del código fuente del proyecto para conexión a base de datos y lógica de la aplicación



Sprint burndown chart



Projects / 533Grupo1 / Reports

Sprint burndown chart

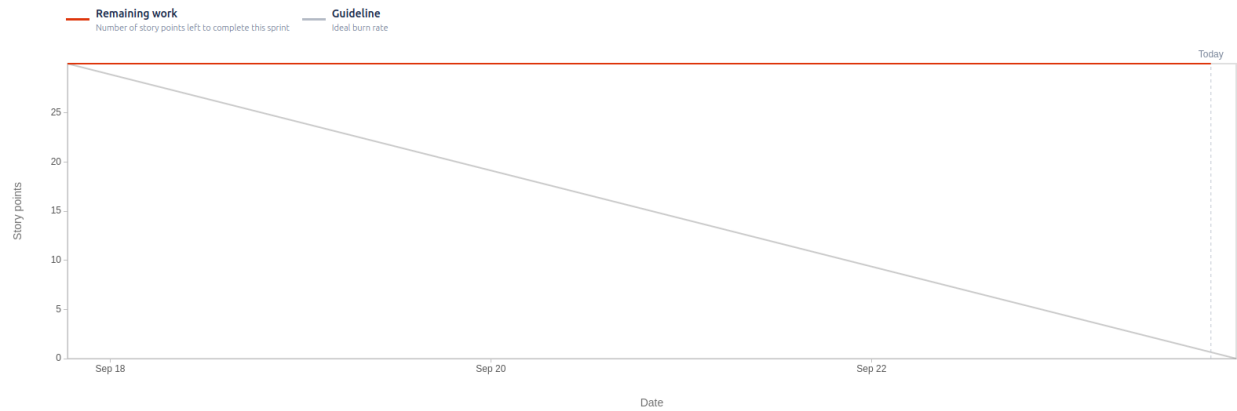
[How to read this](#)

Sprint
Sprint 4

Estimation field
Story points

Date - 17 September 2021 - 23 September 2021

Sprint goal - Entrega del código fuente del proyecto para conexión a base de datos y lógica de la aplicación



Cumulative flow diagram

Projects / 533Grupo1 / Reports

Cumulative flow diagram

[How to read this report](#)

Date filter
All Time

From date
2/18/1993

To date
2/18/1993

☒ TO DO ☒ IN PROGRESS ☒ PEER REVIEW ☒ DONE

