CURSO INTEGRADOR I SISTEMAS - SOFTWARE

Semana 12

Profesor: Roosevelt López

Maven Invocación

```
<project>
    ...
    <dependencies>
        <dependency>
            <groupId>com.oracle.database.jdbc</groupId>
            <artifactId>ojdbc8</artifactId>
            <version>19.8.0.0</version>
        </dependency>
        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-context</artifactId>
            <version>5.3.8</version>
        </dependency>
        ...
    </dependencies>
    ...
</project>
```

Gradle

```
apply plugin: 'java’

Repositories
 { mavenCentral() }

dependencies {
compile group: 'com.oracle.database.jdbc', name: 'ojdbc8', version: '19.8.0.0’
compile group: 'org.springframework', name: 'spring-context', version: '5.3.8’
 }
```

CONTENIDO DE LA SESIÓN

1. Manejo de Datos
2. Control de Versiones

UTILIDAD

- En que nos puede servir control de versiones

Logro

- Al finalizar la sesión, el estudiante podrá reforzar los conceptos de manejo de datos y control de versiones.

UNIVERSIDAD TECNOLÓGICA DEL PERÚ

UNIDAD 05

Coordinación de Proyecto

Semana 12

Conexión base de Datos

```java
package JDBC_ORACLE_01;
import java.sql.*;

class OracleCon{

    public static void main(String args[]){
    try{
      Connection con=DriverManager.getConnection("jdbc:oracle:thin:@localhost:1521:orcl","SYSTEM","123456utP");
      Statement stmt=con.createStatement();

      ResultSet rs=stmt.executeQuery("select * from alumnos");
      while(rs.next()) {
         System.out.println(rs.getInt(1)+"  "+rs.getString(2)+"  "+rs.getString(3));
      }
      con.close();

    }catch(SQLException e){
        System.out.println(e);}
    }
}
```

Conexión a base de Datos

```java
package JDBC_ORACLE_02;
import java.sql.*;
import oracle.jdbc.driver.OracleDriver;


public class ManejoBaseDatos {

    private String url;

    private String usuario;

    private String password;

    public ManejoBaseDatos(){
       url="jdbc:oracle:thin:@localhost:1521:orcl";
       usuario="SYSTEM";
       password="123456utP";
    }

    public void listarAlumnos(){
        try{
         Connection cn=DriverManager.getConnection(url,usuario,password);
         Statement obj = cn.createStatement();
         ResultSet rs = obj.executeQuery("select * from alumnos");
         while(rs.next()){
         System.out.println("Identificador: "+rs.getInt(1));
         System.out.println("Nombre: "+rs.getString(2));
         System.out.println("Pais: "+rs.getString(3));
         System.out.println("Fecha de Nacimiento: "+rs.getDate(4));
         System.out.println("Sexo: "+rs.getString(5));
         System.out.println("Curso: "+rs.getString(6));
         System.out.println();
         }
        }
        catch(SQLException e){
          System.out.println("Error: "+e.getMessage());
        }
    }
}
```

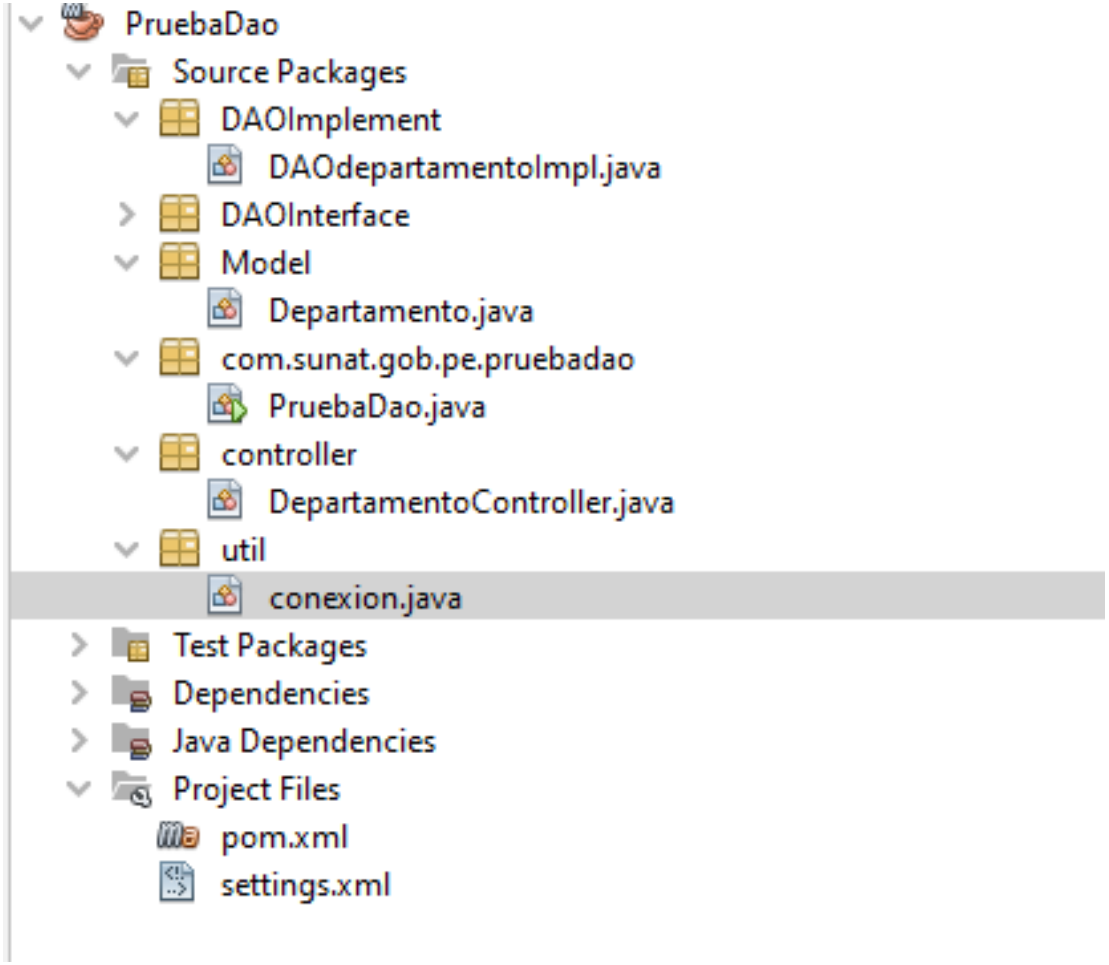
```java
package JDBC_ORACLE_02;

public class Prueba {

    public static void main(String args[]){
      ManejoBaseDatos m = new ManejoBaseDatos();
      m.listarAlumnos();
    }
}
```

Conexión a BD: Procedure

```
JDK is not defined
package JDBC_ORACLE_03;

public class Prueba {

    public static void main(String args[]){
      ManejoBaseDatos m = new ManejoBaseDatos();
      m.listarAlumnosUsandoProcedimiento();
    }
}
```

```
public class ManejoBaseDatos {

    private  String url;

    private  String usuario;

    private  String password;

    public ManejoBaseDatos(){
       url="jdbc:oracle:thin:@localhost:1521:orcl";
       usuario="SYSTEM";
       password="123456utP";
    }

    public void listarAlumnosUsandoProcedimiento(){
        try{
         Connection cn=DriverManager.getConnection(url,usuario,password);
         CallableStatement obj = cn.prepareCall("{call listado(?)}");
         obj.registerOutParameter(1,OracleTypes.CURSOR);
         obj.execute();

         ResultSet rs=(ResultSet)obj.getObject(1); // EL 1 HACE REFERENCIA ÚNICO PARÁMETRO, ES DECIR EL CURSOR.

         while(rs.next()){
         System.out.println("Identificador: "+rs.getInt(1));
         System.out.println("Nombre: "+rs.getString(2));
         System.out.println("Pais: "+rs.getString(3));
         System.out.println("Fecha de Nacimiento: "+rs.getDate(4));
         System.out.println("Sexo: "+rs.getString(5));
         System.out.println("Curso: "+rs.getString(6));
         System.out.println();
         }
        }
        catch(SQLException e){
          System.out.println("Error: "+e.getMessage());
        }
    }

}
```

CONEXIÓN DATOS DAO



UNIVERSIDAD TECNOLÓGICA DEL PERÚ

Conexion

```
public class conexion {
    private Connection conn;

    public Connection getConexion() {

        try {
            Class.forName( className: "com.mysql.cj.jdbc.Driver");
            conn = DriverManager.getConnection( url: "jdbc:mysql://localhost:3306/bdairbnfx",  user: "bdairbnFX",  password: "123456");

        } catch (ClassNotFoundException | SQLException e) {
            System.out.println( x: e.getMessage());
        }

        return conn;
    }
}
```

Llamada de DAO

```
public class DAOdepartamentoImpl implements departamentoDAO{

    @Override
    public List<Departamento> listardepartamento() {
       conexion conexionC = new conexion();
        Connection conn = conexionC.getConexion();
        PreparedStatement pstmt = null;
        List<Departamento> listaDepartamento = new ArrayList<>();
        ResultSet rs = null;
        try {
            String sql = "select  iddepartamento,Distrito, Direccion,PrecioNoche   from departamento";
            pstmt = conn.prepareStatement(sql);

            rs = pstmt.executeQuery();
            while (rs.next()) {
                listaDepartamento.add(new Departamento( iddepa: rs.getInt( columnIndex: 1),  distrito: rs.getString( columnIndex: 2),  direccion: rs.getString( columnIndex: 3), precioNoche: rs.getDouble( columnIndex: 
                 //   System.out.println("Distrito:" + rs.getString(2));
            }
        } catch (SQLException se) {
            System.out.println( x: se.getMessage());
        } finally {
            try {
                if (conn != null) {
                    conn.close();
                }
                if (pstmt != null) {
                    pstmt.close();
                }
                if (rs != null) {
                    rs.close();
                }
            } catch (SQLException se) {
                System.out.println( x: se.getMessage());
            }
        }
        return listaDepartamento;
    }

}
```

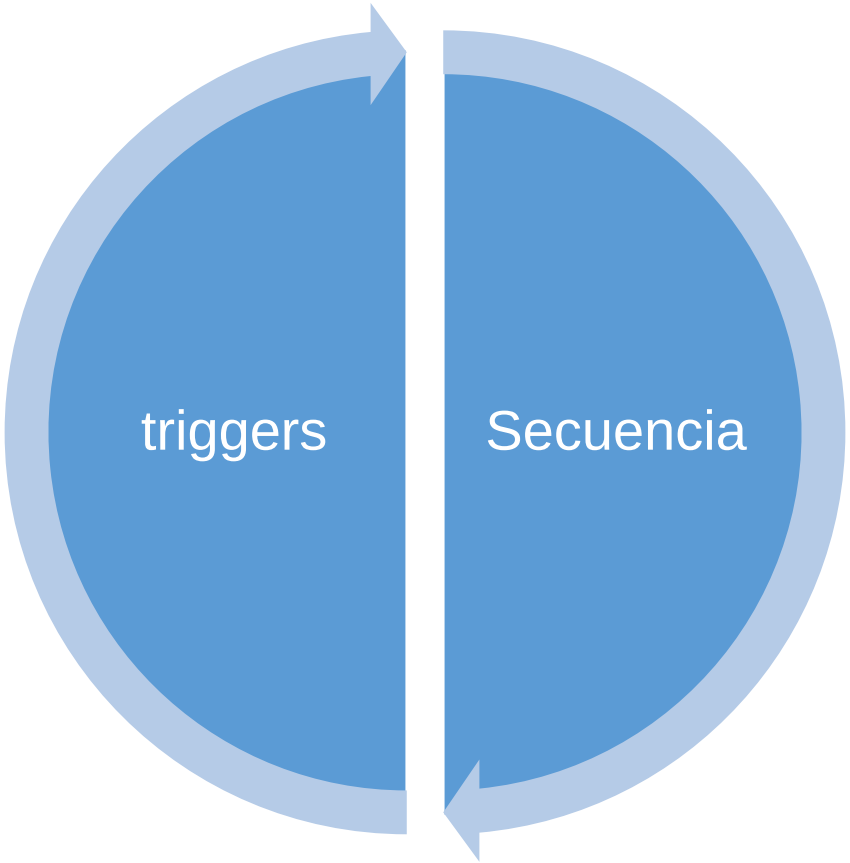
CONTROLLER

```java
package controller;

import DAOImplement.DAOdepartamentoImpl;
import DAOInterface.departamentoDAO;
import Model.Departamento;
import java.util.List;

public class DepartamentoController {

    public    List<Departamento> devolverListaDepartamento(){

     departamentoDAO depa=new DAOdepartamentoImpl();

     List<Departamento> listaDepartamento = depa.listardepartamento();
     return listaDepartamento;
    }

}
```

Ejecución de MAIN

```java
package com.sunat.gob.pe.pruebadao;

import Model.Departamento;
import controller.DepartamentoController;
import java.util.List;

public class PruebaDao {

    public static void main(String[] args) {

        DepartamentoController depacontroller=new DepartamentoController();

    //   System.out.println("Hello World!");
     List<Departamento> listaDepartamento = depacontroller.devolverListaDepartamento();

    for(Departamento depto : listaDepartamento) {
    System.out.println("ID: " + depto.getIddepa() + ", Nombre: " + depto.getDireccion()+ ", PRECIO: " + depto.getPrecioNoche()) ;
        }
    }
}
```

Objetos de Base de Datos



Secuencia

```
CREATE SEQUENCE SEAlumno
  START WITH 1
  MAXVALUE 9999999
  NOCYCLE
  CACHE 20;
```

- La secuencia comienza con 1 (**START WITH 1**)
- El valor máximo que puede generar es 9999999 (**MAXVALUE 9999999**).
- La opción **NOCYCLE** significa que la secuencia no volverá a empezar desde 1 cuando llegue al valor máximo.
- **CACHE 20** significa que Oracle pregenerará 20 números de secuencia para un acceso más rápido.

```
SELECT SEAlumno.NEXTVAL FROM DUAL;
```

Triggers Insert

```
-- Trigger para INSERT
CREATE OR REPLACE TRIGGER T01I_ALUMNO
BEFORE INSERT ON Alumno
FOR EACH ROW
BEGIN
    :new.UsuCreacion := USER;
    :new.FecCreacion := SYSDATE;
    :new.UsuModificacion := USER;
    :new.FecModificacion := SYSDATE;
END;
/
```

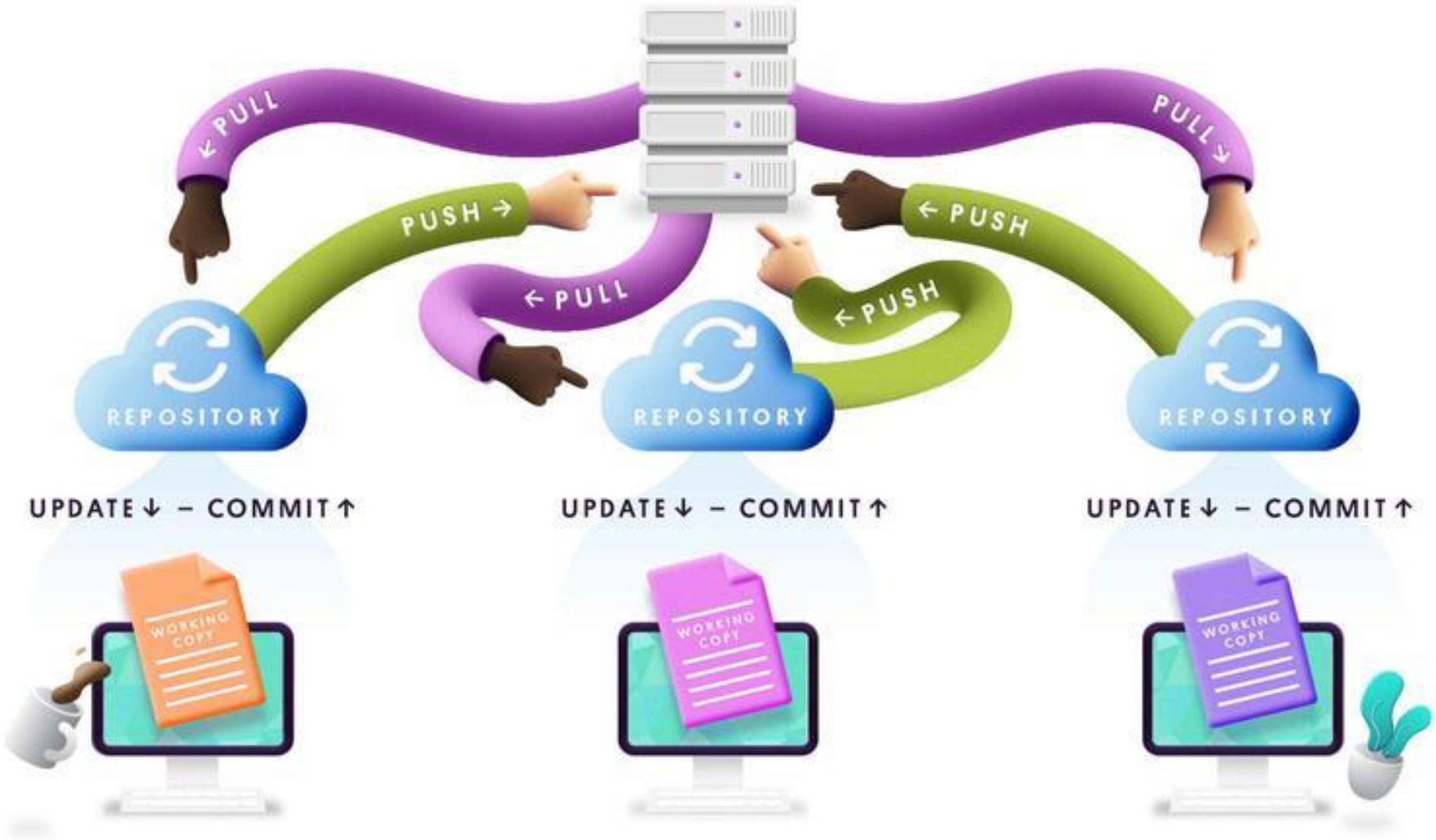
Triggers Update

```
-- Trigger para UPDATE
CREATE OR REPLACE TRIGGER T02U_ALUMNO
BEFORE UPDATE ON Alumno
FOR EACH ROW
BEGIN
    :new.UsuModificacion := USER;
    :new.FecModificacion := SYSDATE;
END;
/
```

Trigger Combinado

```
-- Trigger para INSERT y UPDATE
CREATE OR REPLACE TRIGGER T03IU_ALUMNO
BEFORE INSERT OR UPDATE ON Alumno
FOR EACH ROW
BEGIN
   IF INSERTING THEN
      :new.UsuCreacion := USER;
      :new.FecCreacion := SYSDATE;
     :new.UsuModificacion := USER;
     :new.FecModificacion := SYSDATE;
   END IF;
   :new.UsuModificacion := USER;
   :new.FecModificacion := SYSDATE;
END;
/
```

Control de Versiones

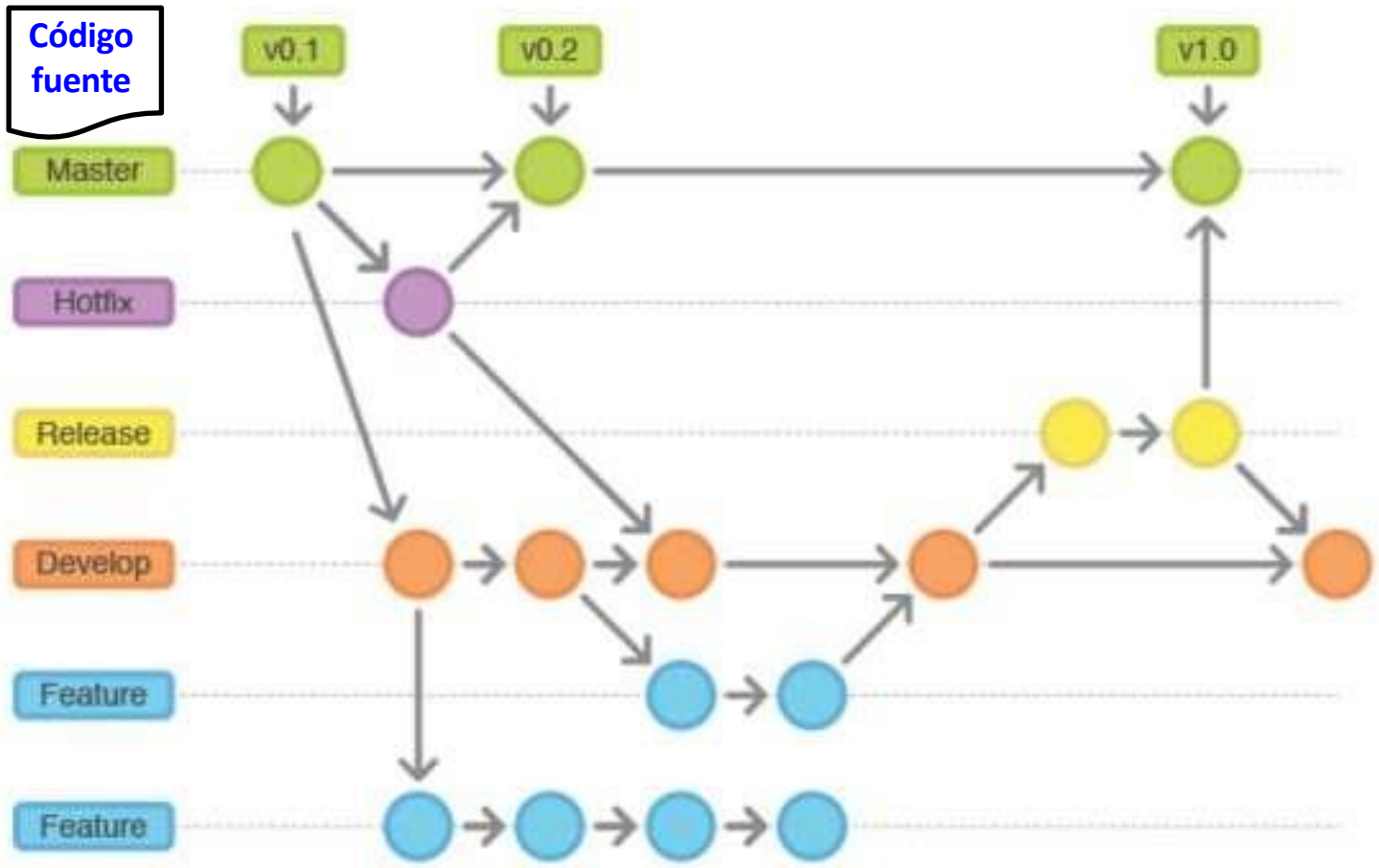


¿Qué es GIT?

Es un software para guardar los cambios en los archivos y sub-carpetas dentro de una carpeta principal, como por ejemplo un código de un aplicativo.

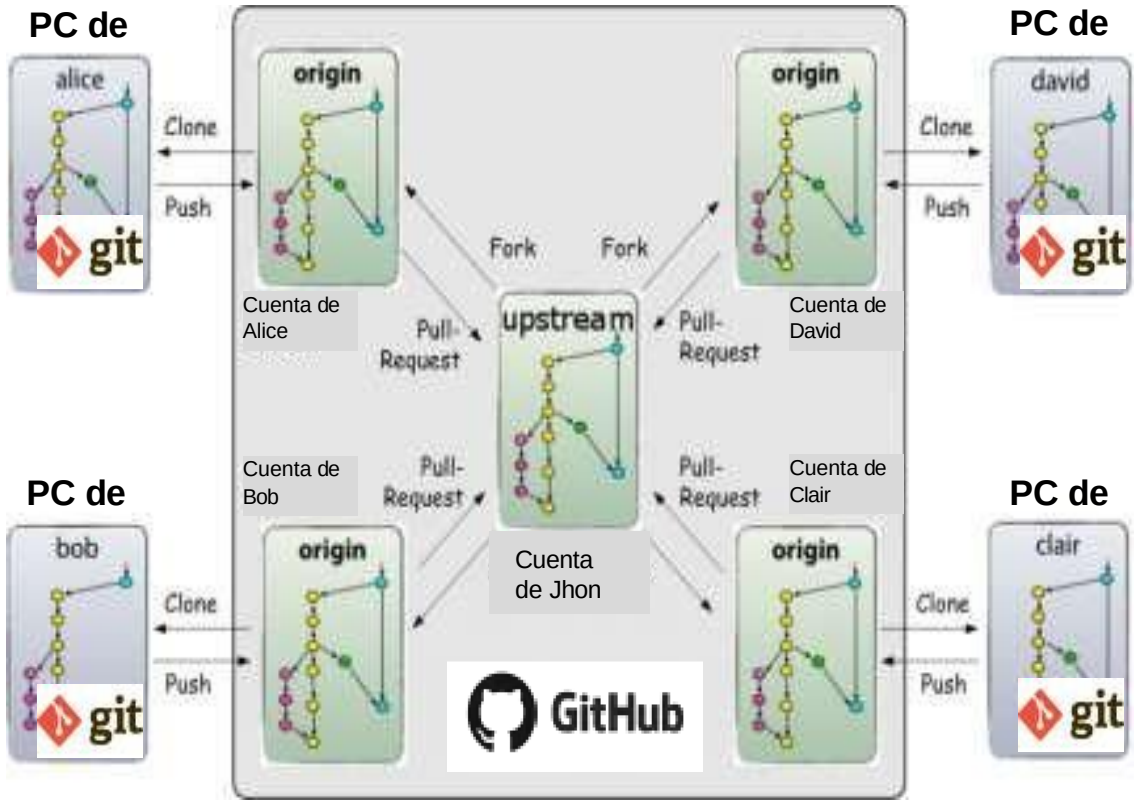
Además, permite tener muchas copias independientes del código original, lo que facilita hacer arreglos (**hotfix**), o nuevas opciones (**feature**) sin afectar el código principal (**master**).

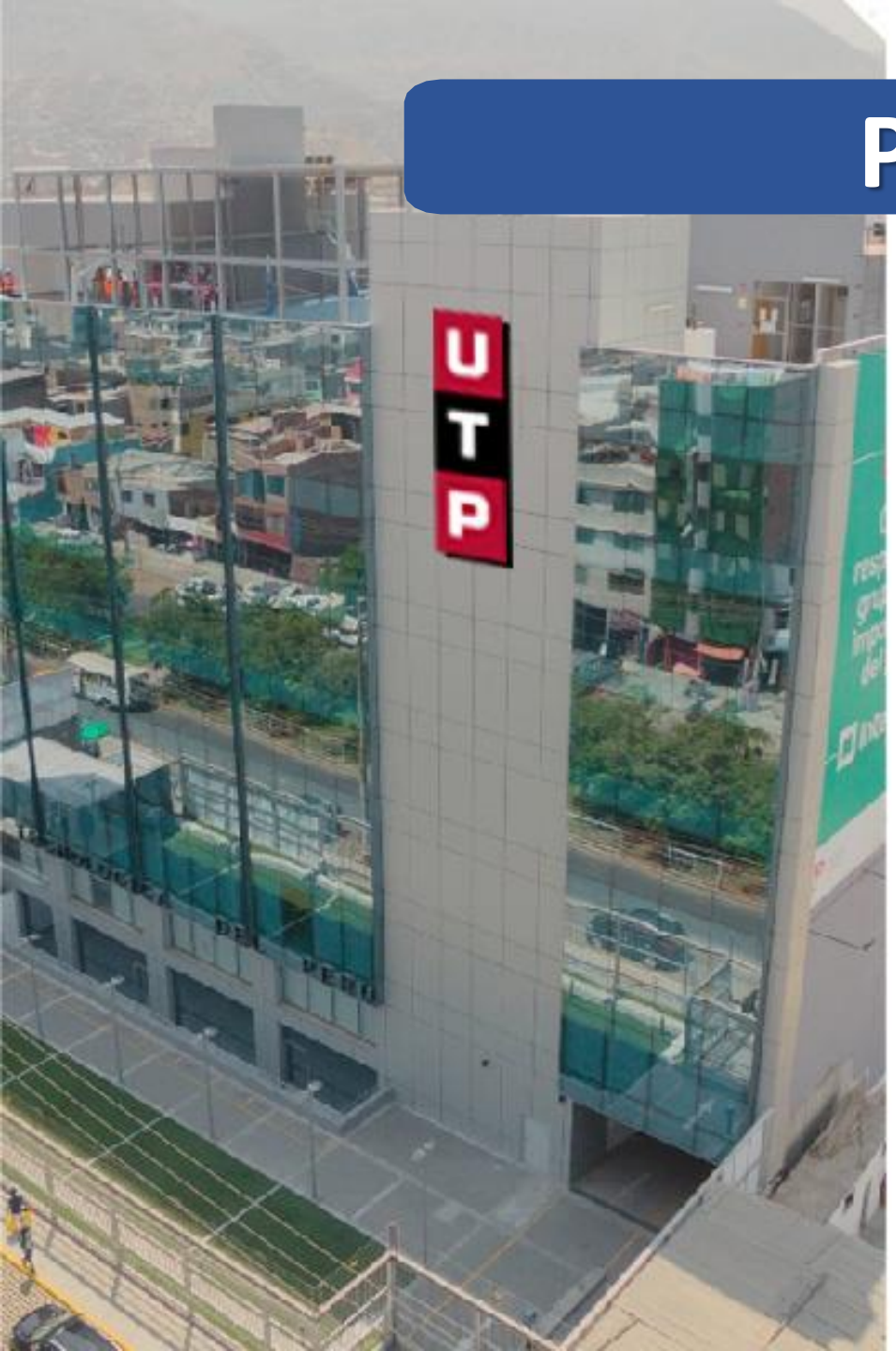
Los cambios (**release**) aprobados posteriormente se pueden integrar al código principal (**master v1.0**) para crear las nuevas versiones



¿Qué es GitHub?

- Git guarda las versiones en tu PC, pero para almacenarlas en la nube requiere un servicio como GitHub que te permite crear una cuenta gratuita. GitHub se basa en Git para el control de versiones, pero ofrece otras facilidades como poder documentar el proyecto (**wiki**), seguimiento de errores (**issues**), control del proyecto (**projects**), copiar otro proyecto a tu cuenta (**fork**), aporte de otros usuarios (**pull request**), y automatización para compilación, pruebas, e implementación en un servidor, es decir, devops (**actions**).

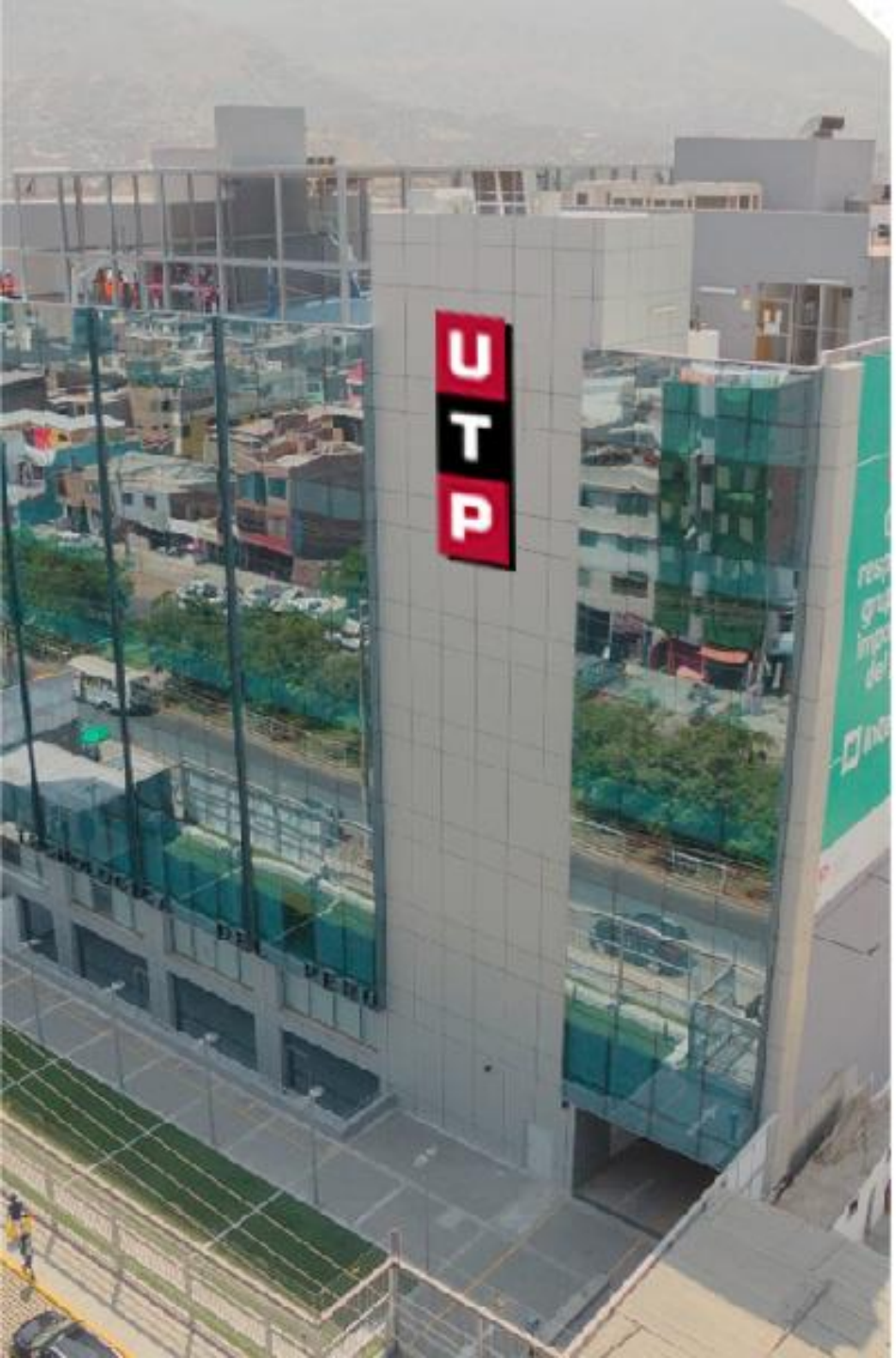




Preguntas

Campus
San Juan de Lurigancho

Universidad Tecnológica del Perú



Incluir los Campos de Auditorias que se actualizara por Triggers

Controlar su proyecto por medio del Git (Indicar el repositorio GITHUB de su proyecto)

Campus
San Juan de Lurigancho

Universidad Tecnológica del Perú


UTP
Universidad
Tecnológica
del Perú