**BASES DE DATOS**

**PROYECTO ENTREGA 3**

[Diego Nicolas Ramirez Maldonado](mailto:dieramirezma@unal.edu.co)

[Axel Gomez Moreno](mailto:axgomezm@unal.edu.co)

[Felipe Santiago Valderrama Ballesteros](mailto:fvalderramab@unal.edu.co)

**1 Hacer un trigger para actualizar otras tablas (solicitudes de facultad y asi) con cada inserción que se haga (cada vez que un estudiante inserte una solicitud que el admin ponga el procedimiento estudiante\_crear\_solicitud)**

**2 Triggers para revisar tipo de datos?**

**PROCEDIMIENTOS ALMACENADOS:**

**DELIMITER $$**

**CREATE PROCEDURE verificar\_estudiante(IN cedulaEstudiante INT, OUT nivel VARCHAR(30))**

**BEGIN**

**IF EXISTS (SELECT 1 FROM pregradoest WHERE perCedulaEst = cedulaEstudiante)**

**THEN SET nivel = 'Pregrado';**

**ELSEIF EXISTS(SELECT 1 FROM posgradoest WHERE perCedulaEst = cedulaEstudiante)**

**THEN SET nivel = 'Posgrado';**

**ELSE**

**SET nivel = '0';**

**END IF;**

**END $$**

**DELIMITER;**

Este procedimiento se encarga de verificar el nivel de estudios que cursa un estudiante buscando por su número de cédula. Puede ser pregrado, posgrado o 0, en caso de no estar en ninguno de los anteriores.

**DELIMITER $$**

**CREATE PROCEDURE obtener\_facultad\_id(IN cedulaEstudiante INT, OUT faultadID INT)**

**BEGIN**

**SELECT f.facId INTO faultadID**

**FROM facultad f**

**LEFT JOIN carrera c ON f.facId = c.facId**

**LEFT JOIN estudiapregrado ep ON c.carSNIES = ep.carSNIESPregrado**

**LEFT JOIN estudiaposgrado epos ON c.carSNIES = epos.carSNIESPosgrado**

**WHERE ep.perCedulaEst = cedulaEstudiante OR epos.perCedulaEst = cedulaEstudiante;**

**END $$**

**DELIMITER;**

Procedimiento que se encarga de obtener el ID de la facultad a la que se encuentra asociado un estudiante, a partir de su número de cédula.



**DELIMITER $$**

**CREATE PROCEDURE solicitudes(**

**IN cedulaEstudiante INT,**

**IN iDNomSolicitud INT,**

**IN medio VARCHAR(45),**

**IN tipo VARCHAR(45),**

**IN comentarios LONGTEXT,**

**OUT lastSolicitud INT**

**)**

**BEGIN**

**INSERT INTO solicitud (solMedio, solTipo, estSolFechaEnvio, estSolComentarios, estSolEstadoSolicitud)**

**VALUES (medio, tipo, CURRENT\_DATE(), comentarios, 'Sin resolver');**

**SET @ultimaSolId = LAST\_INSERT\_ID();**

**INSERT INTO estudiante\_has\_solicitud (perCedulaEstudiante, solId)**

**VALUES (cedulaEstudiante, @ultimaSolId);**

**SET lastSolicitud = @ultimaSolId;**

**END $$**

**DELIMITER;**

Procedimiento que permite insertar una solicitud en esta tabla y que retorna el id autoincremental que esta genera. Este procedimiento es útil para poder realizar la relación entre el estudiante y la solicitud que este crea.



**DELIMITER $$**

**CREATE PROCEDURE solicitud\_facultad(**

**IN cedulaEstudiante INT,**

**IN lastSolId INT,**

**IN iDNomSolicitud INT,**

**IN facultadSolTipo VARCHAR(45)**

**)**

**BEGIN**

**CALL obtener\_facultad\_id(cedulaEstudiante, @facID);**

**INSERT INTO facultadsolicitud (solId, nomId, conIdFacultad, facsolTipo)**

**VALUES (lastSolId,iDNomSolicitud, @facID,facultadSolTipo);**

**END $$**

**DELIMITER;**

Procedimiento para insertar en la tabla facultadSolicitud la solicitud que el estudiante cree, dependiendo de la facultad a la que pertenezca, para esto, se usa uno de los PAS anteriores.



**DELIMITER $$**

**CREATE PROCEDURE solicitud\_sede(**

**IN lastSolId INT,**

**IN iDNomSolicitud INT**

**)**

**BEGIN**

**INSERT INTO sedesolicitud (solId, conIdSede, nomId)**

**VALUES (lastSolId,1,iDNomSolicitud);**

**END $$**

**DELIMITER;**

Procedimiento para insertar una solicitud en la tabla sedeSolicitud



**DELIMITER $$**

**CREATE PROCEDURE estudiante\_crear\_solicitud(**

**IN cedulaEstudiante INT,**

**IN iDNomSolicitud INT,**

**IN medio varchar(45),**

**IN comentarios LONGTEXT**

**)**

**BEGIN**

**CALL verificar\_estudiante(cedulaEstudiante, @nivel);**

**CASE @nivel**

**WHEN 'Pregrado' THEN**

**IF(iDNomSolicitud>0 AND iDNomSolicitud<21)THEN**

**CALL solicitudes(cedulaEstudiante, iDNomSolicitud, medio, 'Facultad' ,comentarios, @ultimaSolId);**

**CALL solicitud\_facultad(cedulaEstudiante,@ultimaSolId,iDNomSolicitud, 'Solicitud de pregrado');**

**ELSEIF(iDNomSolicitud>26 AND iDNomSolicitud<37)THEN**

**CALL solicitudes(cedulaEstudiante, iDNomSolicitud, medio, 'Sede' ,comentarios, @ultimaSolId);**

**CALL solicitud\_sede(@ultimaSolId,iDNomSolicitud);**

**ELSE select'solicitud no permitida';**

**END IF;**

**WHEN 'Posgrado' THEN**

**IF(iDNomSolicitud>0 AND iDNomSolicitud<23)THEN**

**CALL solicitudes(cedulaEstudiante, iDNomSolicitud, medio, 'Facultad' ,comentarios, @ultimaSolId);**

**CALL solicitud\_facultad(cedulaEstudiante,@ultimaSolId,iDNomSolicitud, 'Solicitud de posgrado');**

**ELSEIF(iDNomSolicitud>22 AND iDNomSolicitud<27)THEN**

**CALL solicitudes(cedulaEstudiante, iDNomSolicitud, medio, 'Facultad' ,comentarios, @ultimaSolId);**

**CALL solicitud\_facultad(cedulaEstudiante,@ultimaSolId,iDNomSolicitud, 'Solicitud por comité de posgrado');**

**ELSEIF(iDNomSolicitud>26 AND iDNomSolicitud<37)THEN**

**CALL solicitudes(cedulaEstudiante, iDNomSolicitud, medio, 'Sede' ,comentarios, @ultimaSolId);**

**CALL solicitud\_sede(@ultimaSolId,iDNomSolicitud);**

**ELSE select'solicitud no permitida';**

**END IF;**

**ELSE**

**SELECT 'No se encontro un estudiante con esa cedula.';**

**END CASE;**

**END $$**

**DELIMITER ;**

Este procedimiento se encarga de crear una solicitud, se le pasan por parametros valores tales como la cédula del estudiante y los datos básicos de la solicitud que desee crear. Se hace llamado al PA solicitudes para crearla dependiendo de parámetros como si el tipo de solicitud es de sede o facultad, entre otros filtros.



**DELIMITER $$**

**CREATE PROCEDURE estudiante\_ver\_solicitud(**

**IN cedulaEstudiante INT**

**)**

**BEGIN**

**IF EXISTS (SELECT \* FROM estudiante WHERE EXISTS(**

**SELECT 1 FROM estudiante WHERE perCedula = cedulaEstudiante)**

**AND NOT EXISTS(SELECT 1 FROM egresado WHERE perCedulaEst = cedulaEstudiante))THEN**

**SELECT \* FROM solicitud NATURAL JOIN estudiante\_has\_solicitud WHERE cedulaEstudiante = perCedulaEstudiante;**

**ELSE**

**SELECT 'No se encontro un estudiantecon esta cedula';**

**END IF;**

**END $$**

**DELIMITER ;**

Procedimiento para que el estudiante pueda ver las solicitudes que este ha generado. Se utiliza la cédula para filtrar únicamente aquellas que correspondan al estudiante



**DELIMITER $$**

**CREATE PROCEDURE update\_estudiante(**

**IN cedulaEstudiante INT,**

**IN solicitudID INT,**

**IN comentariosNuevos longtext**

**)**

**BEGIN**

**UPDATE solicitud AS s**

**INNER JOIN estudiante\_has\_solicitud AS es ON s.solId = es.solId**

**SET s.estSolComentarios = comentariosNuevos**

**WHERE es.perCedulaEstudiante = cedulaEstudiante AND s.solId = solicitudID;**

**END $$**

**DELIMITER ;**

Procedimiento para actualizar el campo de comentarios en la tabla solicitud usando la cedula del estudiante



**DELIMITER $$**

**CREATE PROCEDURE update\_comite(**

**IN cedulaEstudiante INT,**

**IN solicitudID INT,**

**IN solicitudEstado varchar(45)**

**)**

**BEGIN**

**UPDATE solicitud AS s**

**INNER JOIN estudiante\_has\_solicitud AS es ON s.solId = es.solId**

**SET s.estSolEstadoSolicitud = solicitudEstado**

**WHERE es.perCedulaEstudiante = cedulaEstudiante AND s.solId = solicitudID;**

**END $$**

**DELIMITER ;**

Procedimiento para actualizar el estado de la solicitud, dependiendo de las decisiones tomadas en los comités. Este procedimiento únicamente será ejecutado por los usuarios con el rol de comite

**INDICES**



**CREATE UNIQUE INDEX idx\_estudiantes\_pregrado ON estudiapregrado(perCedulaEst);**

Indice creado sobre la columna perCedulaEst, en la tabla estudiapregrado, para agilizar la búsqueda de los estudiantes que cursan un pregrado.



**CREATE UNIQUE INDEX idx\_estudiantes\_posgrados ON estudiaposgrado(perCedulaEst);**

Indice creado sobre la columna perCedulaEst, en la tabla estudiaposgrado, para agilizar la búsqueda de los estudiantes que cursan un posgrado.

1. **CREATE INDEX idx\_estudiantes\_solicitudes ON estudiante\_has\_solicitud(perCedulaEstudiante);**

Indice creado sobre la columna perCedulaEst, en la tabla estudiante\_has\_solicitud, para agilizar la búsqueda de los estudiantes que han realizado alguna solicitud.

1. **CREATE INDEX idx\_facultades\_solicitudes ON facultadsolicitud(conIdFacultad);**

Indice creado sobre la columna conIdFacultad, en la tabla **facultadsolicitud**, para agilizar la búsqueda de las solicitudes que correspondan a facultad

1. **CREATE INDEX idx\_solicitud ON solicitud(solId);**

Indice creado sobre la columna solId, en la tabla **solicitud**, para agilizar la búsqueda de las solicitudes que se hayan creado

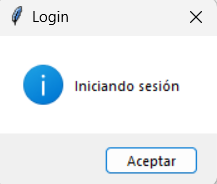
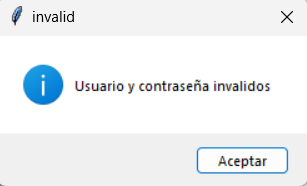
**PROTOTIPO DE LA INTERFAZ**

Para el desarrollo de la interfaz, hemos usando el lenguaje de programación Python con su librería para desarrollo de interfaces gráficas, Tkinter. Haciendo uso de esta herramienta hemos podido implementar funcionalidades basadas en nuestro modelo de base de datos de solicitudes al consejo, para cada rol que hemos establecido, dando permisos, y funcionalidades específicas a cada uno de estos.

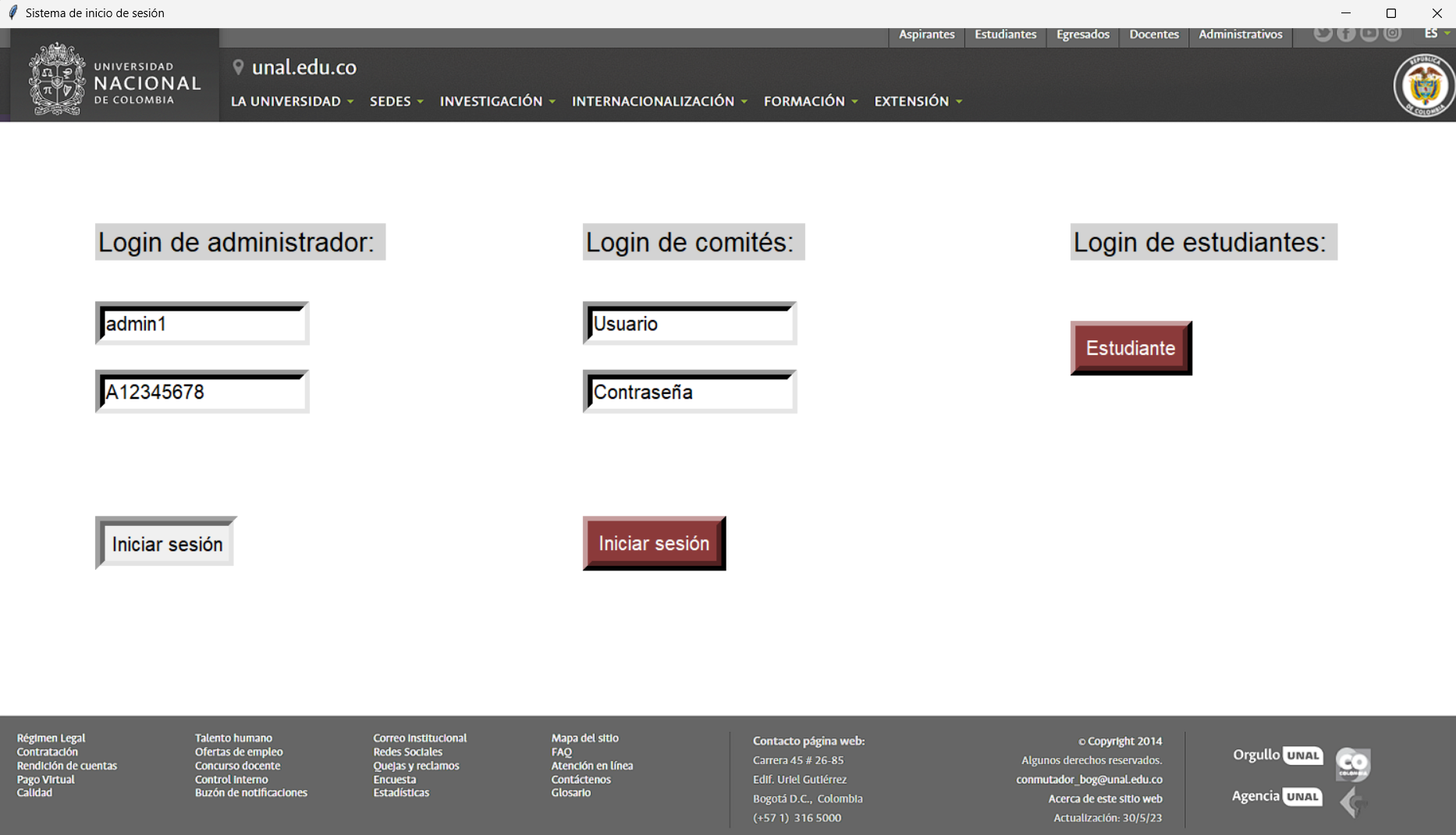
A continuación presentamos algunas capturas de pantalla del desarrollo realizado hasta el momento

inicial.py



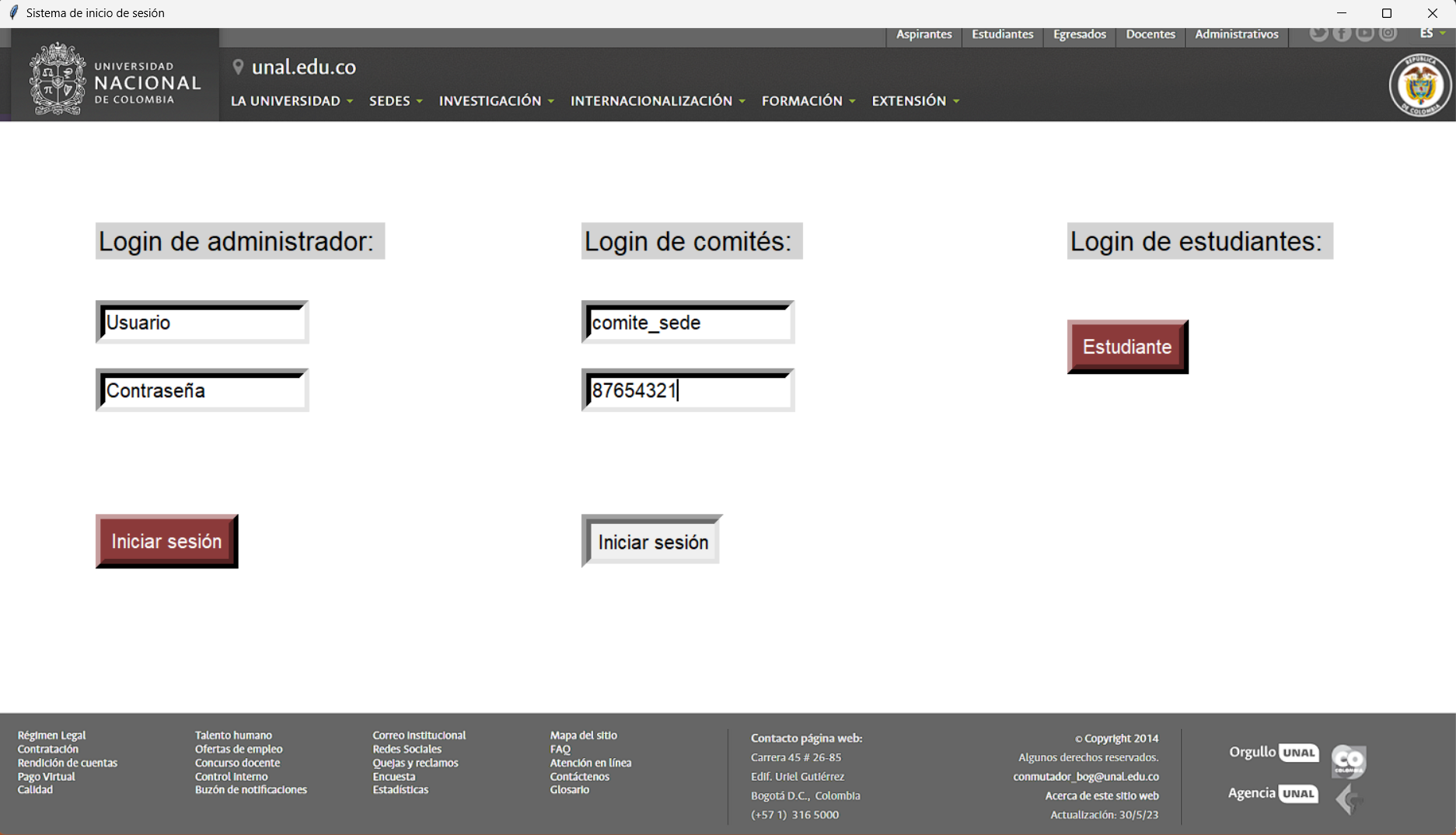
 

Ingreso para administradores (puede insertar solicitudes, eliminarlas, ver la tabla completa de solicitudes, agrupar por cédulas). En este perfil, es posible realizar todos los procedimientos, dado que este rol tiene acceso a toda la DB.

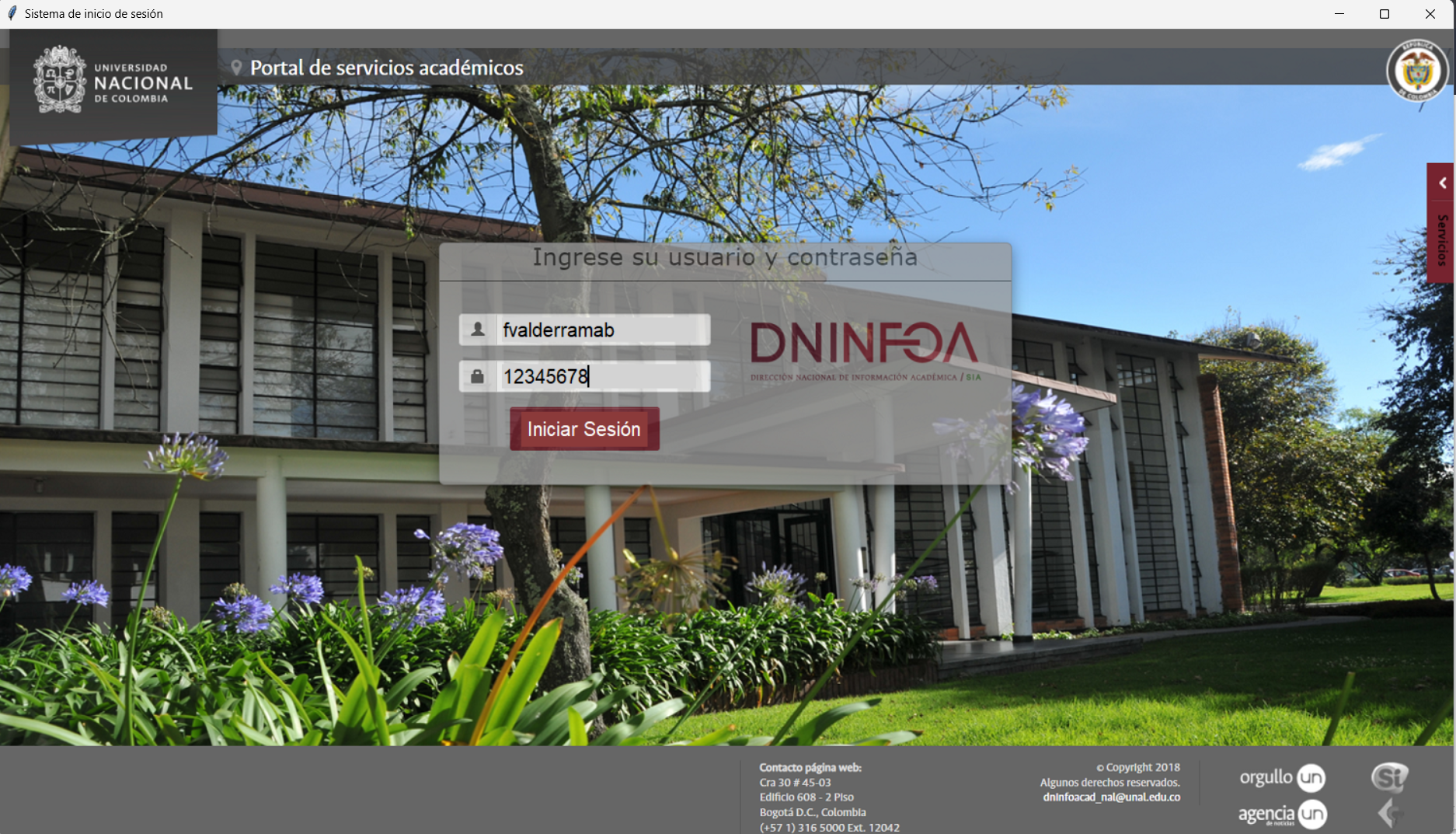
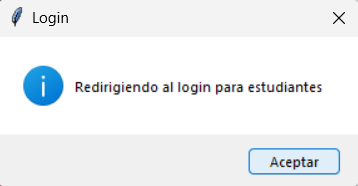
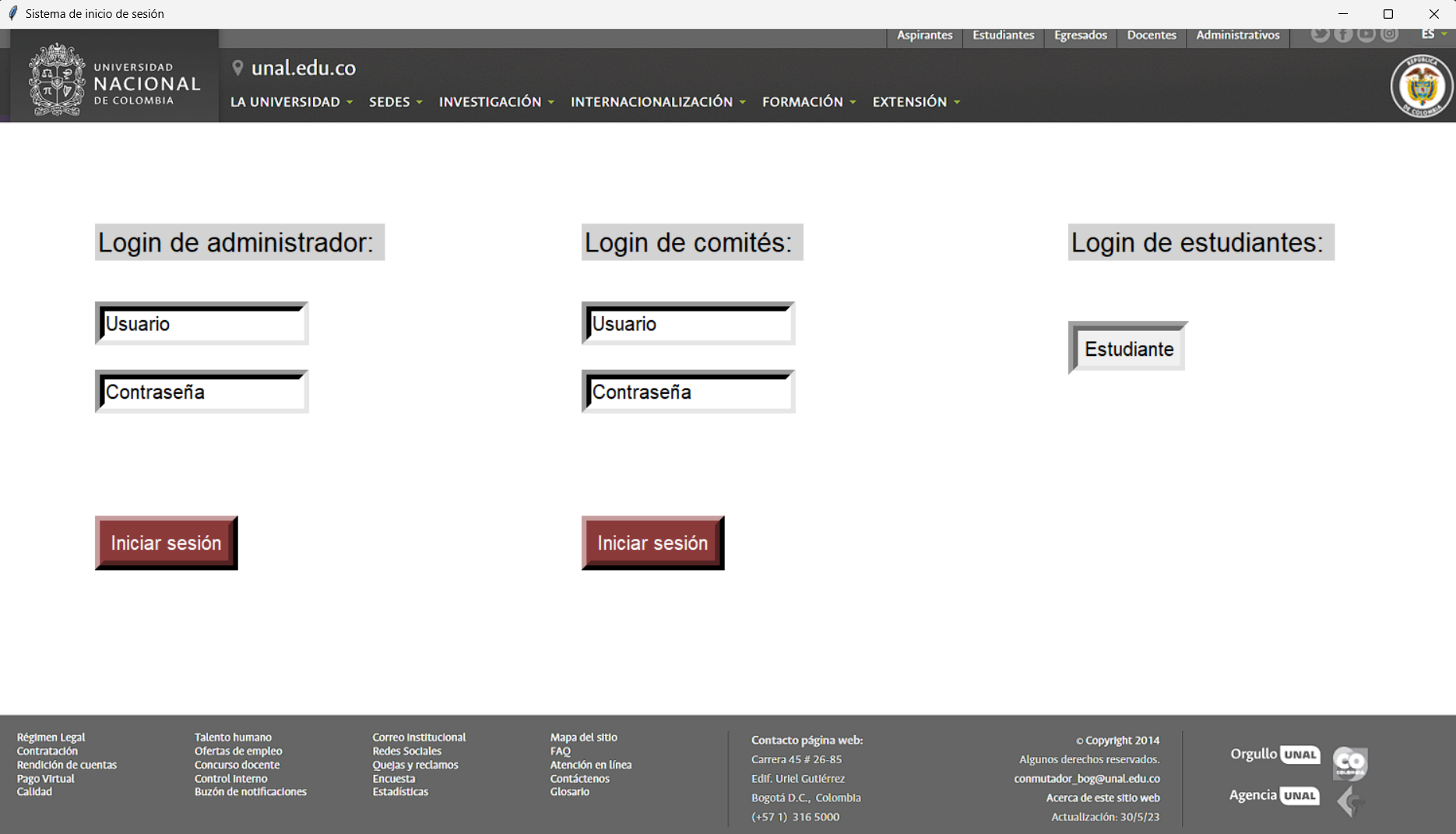




Ingreso para comités (se encargan de evaluar las solicitudes enviadas y resolverlas por lo que cambian el estado de ciertas solicitudes a Resuelta). En este perfil es posible utilizar el procedimiento update\_comite, con el fin de actualizar el estado de las solicitudes hechas por los estudiantes



Ingreso para estudiantes (Solo pueden enviar y consultar solicitudes que corresponden a su cédula). Teniendo en cuenta esto, los procedimientos almacenados que se pueden realizar sobre este rol son estudiante\_Ver\_solicitud y estudiante\_crear\_solicitud, para realizar las acciones mencionadas anteriormente.





**En la carpeta comprimida que se encuentra adjunta en esta entrega, se pueden visualizar los scripts de la DB actualizados, así como el diagrama relacional. También, es posible acceder a los scripts de creación de la interfaz gráfica.**