

work hard & dream big!!!



CAADI

CAADI



DOCUMENTACIÓN

SISTEMA DE REGISTRO DE ASISTENCIAS CAADI

ÍNDICE DE CONTENIDO

INTRODUCCIÓN	1
TRABAJANDO COMO EQUIPOS SEPARADOS	2
TRABAJANDO COMO UN SOLO EQUIPO	20
MANUAL DE USUARIO	23
MANUAL DE ADMINISTRADOR	26

INTRODUCCIÓN

El Centro de Auto Aprendizaje Dirigido (CAADI) es un centro al que los alumnos de la Universidad de Guanajuato sede Yuriria pueden asistir con el objetivo de mejorar su inglés. CAADI cuenta con material de diferentes temáticas. Los alumnos pueden hacer uso de dicho material cuando asisten a CAADI.

La asistencia puede ser de dos maneras: los alumnos pueden ir a CAADI sin necesidad de cubrir algún requisito o bien, si toman la materia de inglés es necesario que asistan una cierta cantidad de horas por semana dependiendo del nivel que se encuentren cursando.

Los profesores necesitan corroborar la cantidad de horas de asistencia de cada uno de los alumnos puesto que en base a eso asignan un porcentaje de su calificación.

Uno de los objetivos del proyecto es llevar a cabo el registro de asistencia de los alumnos y así generar reportes que contengan información necesaria para que los profesores puedan asignar la calificación correspondiente a cada alumno. El otro objetivo del proyecto es tener un registro de todo el material disponible en CAADI.

El propósito de este documento es describir las tecnologías utilizadas en el proyecto de migración del sistema del CAADI de la Universidad de Guanajuato, Campus Irapuato-Salamanca, Sede Yuriria, lo que motivó a realizar la migración de dicho sistema, así como la arquitectura del sistema.

TRABAJANDO COMO EQUIPOS SEPARADOS

El realizar el trabajo por equipos de manera separada, surge por la necesidad de dividir el proyecto en dos partes:

1. Migrar a diferentes tecnologías el Sistema de Registro de Asistencia al CAADI
2. Realizar módulos para registro de material a utilizar en la asistencia al CAADI

El modo de trabajo de los dos equipos fue a base de sprints, siguiendo una metodología ágil. El equipo se proponía tareas, objetivos y metas para una fecha específica. Ambos equipos realizaron de manera separada 2 sprints, y 2 sprints más ya cuando los equipos se unieron a uno solo.

EQUIPO 1

ANÁLISIS DE REQUERIMIENTOS

La lista de requerimientos de la migración del Sistema de Registro de Asistencia:

1. Registro de Alumnos

- a. El alumno ingresa solamente su NUA y aparece mensaje de confirmación si corresponde el nombre del alumno al NUA tecleado para entrar al sistema.
- b. Se ingresa el NUA nuevamente y se muestran las actividades LISTENING, READING, WRITING, VOCABULARY, GRAMMAR para que el alumno de click a la correspondiente que haya realizado.

2. Registrar el tiempo de trabajo por alumno

- a. Realizar el cálculo del tiempo que el alumno estuvo realizando en el CAADI.

3. Listar a los alumnos que se encuentren activos en el CAADI

- a. Mostrar una tabla con los alumnos que se encuentren activos en el CAADI para poder eliminarlos del Sistema de Registro de Asistencia en caso de alguna anomalía

4. Generar reportes de las actividades en formato PDF

- a. Permitir generar reportes con el nombre del alumno, profesor, grupo, periodo y tiempo realizado por el alumno en el CAADI.

DISEÑO DEL SISTEMA

El diseño del Sistema de Registro de Asistencia, trató de seguir la imagen institucional que se muestra en el portal www.alumnos.ugto.mx, con un header y footer similares a los que la Dirección de Tecnologías de la Universidad de Guanajuato utiliza. El cuerpo de la aplicación lógicamente cambiaría. La interfaz inicial del Sistema de Registro de Asistencia cuenta con una caja de texto que permite ingresar el NUA y dos botones, uno de *Ingresar* y otro de *Salir*. Cuando el alumno ingresa su NUA, el sistema debe preguntar si el NUA que se ha ingresado es el correcto para poder iniciar con la sesión. En caso de que el NUA no corresponda al alumno, se habilitan dos botones, *Aceptar* y *Cancelar*. Cuenta con una barra de navegación en la parte superior derecha para visualizar a los alumnos que se encuentran conectados. El sistema permite que el Administrador del sistema, sea capaz de eliminar a usuarios que no se encuentren activos en el CAADI, pero activos en el Sistema de Registro de Asistencia del CAADI.

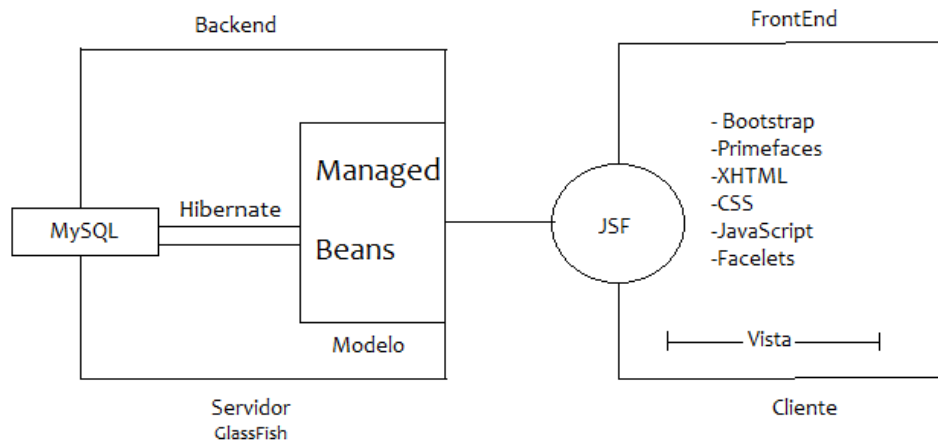
DESARROLLO

Cuando se comenzó la codificación del proyecto, los integrantes del equipo de desarrollo, utilizaron sus cuentas personales de github, así como un repositorio en el que subían sus cambios. La liga al proyecto es: <https://github.com/Arthyom/MigracionCaadi>. El equipo de desarrollo contaba con cuentas personales para entrar a Taiga y visualizar las tareas asignadas para los sprints.

ARQUITECTURA DEL SISTEMA Y TECNOLOGÍAS A UTILIZAR

El equipo 1, durante el primer y segundo sprint, se encargó de desarrollar el registro de asistencia al CAADI. En el desarrollo de la nueva versión del Sistema de Registro de Asistencia del CAADI, se utiliza el Modelo Vista Controlador (MVC) junto con JavaEE. Java Enterprise Edition es el estándar del software empresarial, dirigido por la comunidad Java.

Permite desarrollar y ejecutar software de aplicaciones en el lenguaje de programación Java. La ejecución de las aplicaciones realizadas en Java EE se realizan sobre algún servidor de aplicaciones, normalmente GlassFish, TomCat. Se utilizan las siguientes tecnologías para el desarrollo y un patrón de diseño en automatización:



Modelo Vista Controlador (MVC)

Modelo Vista Controlador (MVC) es un estilo de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos:

- El Modelo contiene una representación de los datos que maneja el sistema, su lógica de negocio y sus mecanismos de persistencia.
- La Vista, o interfaz de usuario, compone la información que se envía al cliente y los mecanismo de interacción con este.
- El controlador actúa como intermediario entre el Modelo y la Vista, gestionando el flujo de información entre ellos y las transformaciones para adaptar los datos a las necesidades de cada uno.

MySQL

MySQL es un sistema de gestión de bases de datos relacional desarrollado por Oracle Corporation. La versión de MySQL que se utiliza es MySQL Community Server 5.7.19. Permite crear base de datos relacionales, gestionar la base de datos.

Hibernate

Hibernate es una herramienta de mapeo objeto-relacional (ORM) para la plataforma Java que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML) o anotaciones en los beans de las entidades que permiten establecer estas relaciones. Hibernate es software libre, distribuido bajo los términos de la licencia GNU LGPL. La versión Hibernate que se utiliza en este desarrollo es 5.1.

Managed Beans

Una aplicación de Java Server Faces típica, incluye uno o más Managed Beans, cada uno puede ser asociado con los componetes usados en una página en particular. Una Managed Beans es una clase Java que sigue la nomenclatura de los JavaBeans. Existen los:

- Beans de Modelo: Representan el Modelo en el patrón MVC.
- Beans de Control: Representan el Controlador en el patrón MVC

Los Managed Beans facilitan la recogida, manipulación y visualización de los valores mostrados en los diferentes elementos de los formularios.

Java Server Faces (JSF)

JavaServer Faces (JSF) es una tecnología y framework para aplicaciones Java basadas en web que simplifica el desarrollo de interfaces de usuario en aplicaciones Java EE. JSF incluye:

- Un conjunto de APIs para representar componentes de una interfaz de usuario y administrar su estado, manejar eventos, validar entrada, definir un esquema de navegación de las páginas y dar soporte para internacionalización y accesibilidad.
- Un modelo de eventos en el lado del servidor.

La versión que se utiliza en esta nueva versión del Sistema de Registro de Asistencias del CAADI Yuriria de la Universidad de Guanajuato es JSF 2.2.

Bootstrap

Bootstrap es un framework o conjunto de herramientas de Código abierto para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús

de navegación y otros elementos de diseño basado en HTML y CSS, así como, extensiones de JavaScript opcionales adicionales. La versión que se utiliza en esta nueva versión del Sistema de Registro de Asistencias del CAADI Yuriria de la Universidad de Guanajuato es v3.0.0.

Primefaces

PrimeFaces es una biblioteca de componentes para JavaServer Faces (JSF) de código abierto que cuenta con un conjunto de componentes enriquecidos que facilitan la creación de las aplicaciones web. Primefaces está bajo la licencia de Apache License V2. Una de las ventajas de utilizar Primefaces, es que permite la integración con otros componentes. Primefaces tiene un conjunto de componentes ricos (Editor de HTML, autocompletar, cartas, gráficas o paneles, entre otros). La versión que se utiliza en esta nueva versión del Sistema de Registro de Asistencias del CAADI Yuriria de la Universidad de Guanajuato es Primefaces 2.

XHTML

XHTML (eXtensible HyperText Markup Language), es básicamente HTML expresado como XML válido. Es más estricto a nivel técnico, pero esto permite que posteriormente sea más fácil al hacer cambios o buscar errores entre otros. La versión XML usada en el proyecto será 1.0.

CSS

CSS es un lenguaje que describe el estilo de un documento html. CSS describe cómo elementos de HTML deben ser mostrados. Es muy usado para establecer el diseño visual de las páginas web, e interfaces de usuario escritas en HTML o XHTML. La versión que se utiliza en esta nueva versión del Sistema de Registro de Asistencias del CAADI Yuriria de la Universidad de Guanajuato es CSS3.

JavaScript

JavaScript (a veces abreviado como JS) es un lenguaje ligero e interpretado, orientado a objetos con funciones de primera clase, más conocido como el lenguaje de script para páginas web, pero también usado en muchos entornos sin navegador, tales como node.js o Apache CouchDB. Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas.

PageObjects

El concepto básico en el que se basa este patrón es el de representar cada una de las pantallas que componen el sitio web o la aplicación que nos interesa probar, como una serie de objetos que encapsulan las características y funcionalidades representadas en la página. De esta manera, nos permite consolidar el código para interactuar con los elementos de una página en cada uno de los PageObjects. Al crear un PageObject, lo que estamos consiguiendo es crear una capa de abstracción entre el " ¿Qué podemos hacer/ver en la página? ", y el " Cómo se realiza esta acción, simplificando enormemente la creación de nuestras pruebas y reutilizando el código con el que interactuamos con la página en concreto. Y a la vez, cualquier cambio que se produzca en la UI únicamente afectará al PageObject en cuestión, no a los test ya implementados.

Facelets

Facelets es un lenguaje de declaración de páginas potente pero ligero que se utiliza para crear vistas de JavaServer Faces mediante plantillas de estilo HTML y para construir árboles de componentes. Las características de Facelets incluyen las siguientes:

- Uso de XHTML para crear páginas web.
- Soporte para bibliotecas de etiquetas de Facelets además de JavaServer Faces y bibliotecas de etiquetas JSTL.

Algunas ventajas de usar Facelets para proyectos de larga escala son las siguientes:

- Soporte para la reutilización de código a través de componentes de plantilla.
- Extensibilidad funcional de componentes y otros objetos del lado del servidor mediante personalización.
- Rápido tiempo de compilación.

En conclusión, el uso de Facelets reduce el tiempo y esfuerzo que necesitas gastar en un desarrollo e implementación de un proyecto.

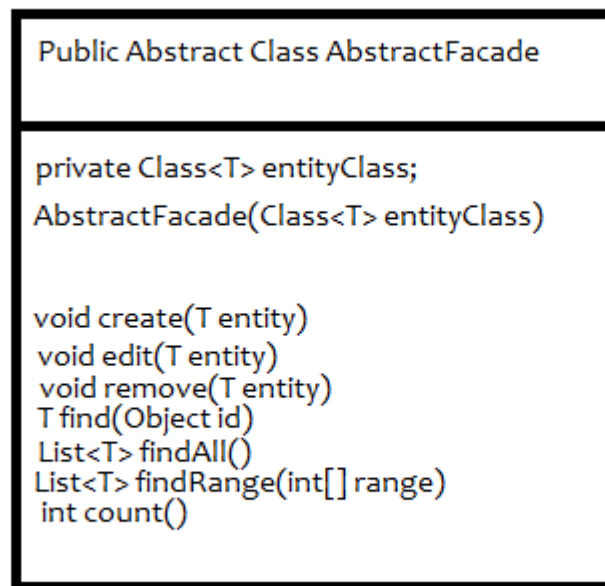
CLASES

Se describen a continuación el funcionamiento del patrón fachada, EJBs, Beans, Entities y JSFs.

PATRÓN FACHADA Y EJBs TIPO FACADE

Los patrones de diseño son soluciones de ingeniería de software a problemas recurrentes en el mundo del desarrollo. Cuando aparece el problema X se suele solventar con el patrón de diseño. El patrón de tipo Facade, es un patrón de diseño que nos permite simplificar el interface de comunicación entre dos objetos A y B de tal forma que para el objeto A sea más sencillo interactuar con el objeto B.

AbstractFacade



Esta clase sirve como base para todas las otras clases en el paquete *“com.CaadiTables.Beans”*, esta clase contiene la lógica compleja que es ocultada a los demás “fachadas” o facades cuando esta clase es heredada/implementada por cualquier otra.

ENTITY JAVA BEAN

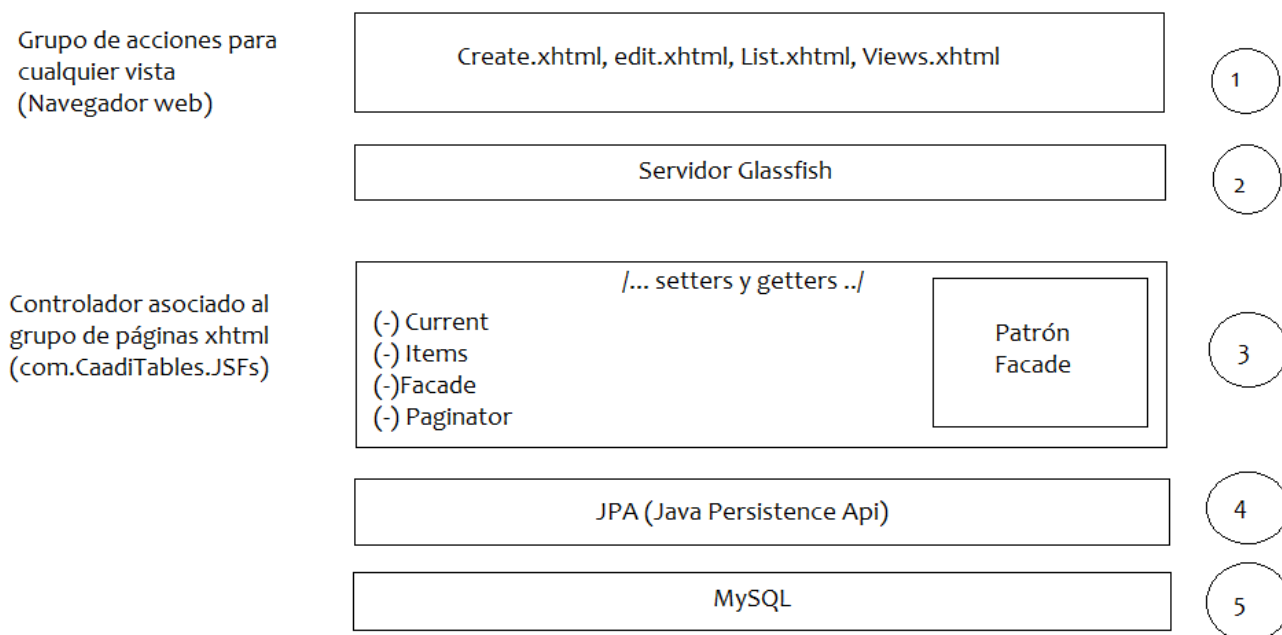
Un Entity Bean es una clase que representa una tabla de una base de datos, y cada instancia de esta clase representa un registro de la tabla, es decir, con los entity beans lo que conseguimos es crear un mapeo entre las propiedades de una clase y los campos de una tabla Además de este mapeo también vamos a poder especificar las relaciones que tienen las clases entre sí (uno a uno, uno a muchos, muchos a uno y muchos a muchos). Todo Entity Bean debe de tener una clave primaria que identifica a ese registro de forma única dentro de la tabla. Todas estas configuraciones las vamos a realizar a

través de anotaciones, y el API que se encarga de gestionar todos los aspectos relativos a la persistencia es JPA (Java Persistent API).

CONTROLADORES (Backing Beans)

Una aplicación típica Java ServerFaces incluye uno o más backing beans, cada uno es un tipo de Java ServerFaces managed bean que puede ser asociado con los componentes usados en una página en particular.

Diagrama de capas Interacción frente-controlador



El nivel 2 del diagrama representa a cualquier clase en el paquete "Com.CaadiTables.JSF". El patrón facade oculta toda la lógica necesaria para conectar con JPA al mismo tiempo estandariza y optimiza la creación de otros controladores para cada grupo de páginas xhtml que está a su vez relacionado con una Entity Java Bean.

HERRAMIENTAS (Clases Java)

Una clase java simple con el único fin de encapsular métodos estáticos con fines diversos, estos fines pueden ser pero no se limitan a:

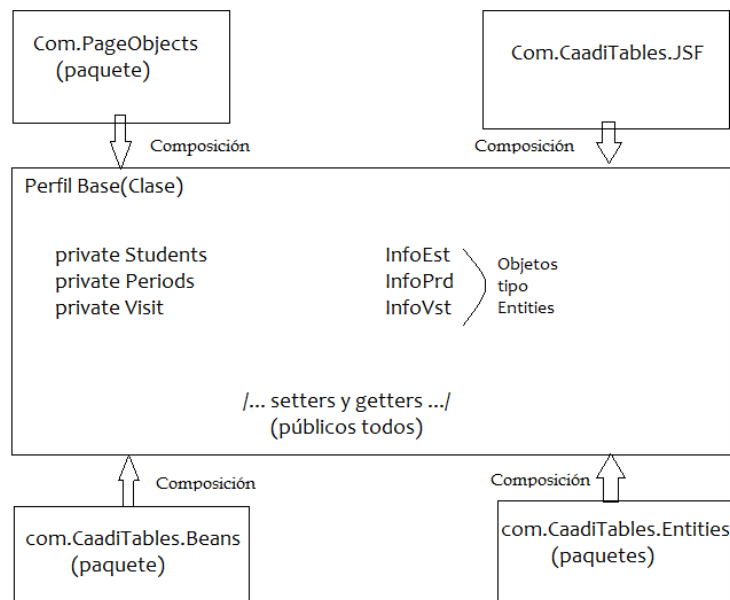
1. Administrar(crear, eliminar, actualizar, insertar, revisar) las sesiones en la aplicación.
2. almacenar objetos para facilitar la comunicación entre el motor de persistencia de la aplicación (JPA), se debe entender por “facilitar” a la centralización y el encapsulamiento de código siguiendo el patrón DRY.

PERFIL BASE (Clase Java)

Una clase simple de java creada con la finalidad de encapsular objetos complejos contruidos a partir de Entity Java Beans. Esta clase almacena instancias de las entities:

1. Students(Tabla estudiantes en la base de datos “CaadiIntegrada”).
2. Teachers(Tabla maestros en la base datos “CaadiIntegrada”).
3. Visit(Tabla visitas en la base datos “CaadiIntegrada”).
4. Periods (Tabla periodos en la base datos “CaadiIntegrada”).

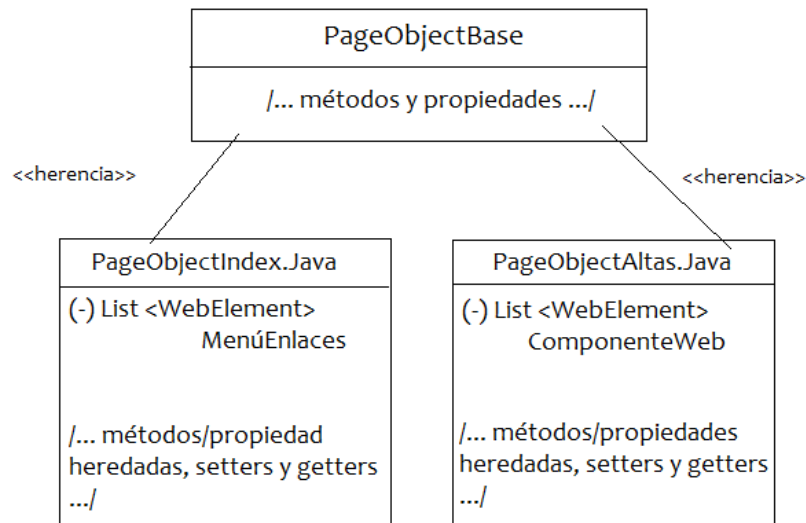
Esta clase es usada principalmente para guardar el estado y la información de una sesión durante el uso de la aplicación web.



Composición es un objeto contenido de otro objeto. Indica que la clase PerfilBase puede ser instanciada por cualquier otra clase en los paquetes del diagrama y viceversa.

PATRÓN PAGEOBJECT

El patrón Page Object nos ayuda con la tarea de robustecer nuestras pruebas. Consiste en crear un objeto por cada conjunto de elementos significativos de la interfaz con la que interactuamos. Aunque el nombre nos sugiera (y generalmente suceda) que cada objeto debe representar una pagina de nuestra aplicación, lo cierto es que si dentro de la pagina tenemos componentes visuales que reutilizamos en otras partes, podemos construir un Page Object de ese elemento para poder reutilizarlo.



Cada página desarrollada debió ser testeada a través de un Page Object...PageObjectIndex, PageObjectAltas, PageObjectBajas, etc. Este enfoque facilita la automatización y el diseño de pruebas.

EVALUACIÓN DE sprint 1 y 2

La evaluación que se tuvo una vez terminados los dos primeros sprints, fue satisfactoria. Se cumplieron de manera correcta la mayoría de las tareas como eran realizar documentación de la arquitectura del sistema, comenzar con el desarrollo de la interfaz inicial y ser capaz de realizar el registro de asistencia del estudiante, realizar algunas pruebas de funcionalidad y registrar tareas en

Taiga. Además de montar el equipo en el CAADI e instalar las herramientas necesarias para desarrollo e implementación. Se tuvo buena comunicación entre los miembros del equipo una vez terminados esos sprint y comenzar la unión con el equipo #2 y su módulo de registro de material.

PROCESO DE INSTALACIÓN Y CONFIGURACIÓN DE EQUIPOS EN EL CAADI

Para ejecutar el Sistema de Registro de Asistencias del CAADI, se dispuso de una iMac de 21.5", con procesador Intel Core i5 dual core de 2.3 GHz. Se ha creado una cuenta de tipo administrador:

- **Usuario:** *SraCaadi*
- **Password:** *SraCaadi2017*

Para poder realizar modificaciones, y ejecutar el sistema, se han instalado algunos IDEs, manejadores de bases de datos, herramientas para poder ejecutar aplicaciones java EE. A continuación se especifican los elementos utilizados:

- JDK 1.8.0
- JRE 1.8.0
- Netbeans IDE 8.2
- MySQL Server 5.0
- MySQL Workbench 6.3 CE
 - **User:** *root*
 - **Password:** *CAADI2017*

PROCESO DE INSTALACIÓN Y CONFIGURACIÓN DE SERVIDOR EN EL CAADI

Para poder ejecutar la aplicación del Sistema de Registro de Asistencia del CAADI, se debe utilizar glassfish 4.1.2 ya que en el desarrollo en Netbeans, se utiliza la versión glassfish 4.1.1, pero para poder subir el WAR de la aplicación, es necesario crear un Connection Pool y un JDBC Resource, porque Oracle, quien se encarga del desarrollo de Glassfish, no permite crear pools ni recursos. Después de visitar varios foros en la comunidad de desarrollo de aplicaciones java EE, el equipo se dió cuenta de ello y se solucionó de la siguiente forma:

Una vez descargado el archivo mysql-connector-java-5.1.44-bin.jar lo agregamos a las siguientes rutas: glassfish4/glassfish/lib Y también en: glassfish4/glassfish/domains/domain1/lib/ext.

1. Primero que hay que hacer iniciar el servidor `./asadmin start-domain` y posteriormente ubicarnos la Glassfish Console con la ruta `localhost:4848`.
2. Después hay que crear una JDBC Connection Pool y un JDBC Resource. Primero el JDBC Connection Pool, hay que dirigirse a `Resources > JDBC > JDBC Connection Pools` le das al botón de **NEW** y en **General Settings** asignar:
 - **Pool Name:** MySQLConnPool
 - **Resource Type:** `javax.sql.DataSource`
 - **Database Drive Vendor:** MySQL
3. Después hay que dar al botón de **NEXT**. Te manda a **NEW JDBC CONNECTION POOL (STEP 2 OF 2)** deslizas hacia abajo y estará la tabla **Additional Properties** ahí vas a agregar a las columnas los siguientes valores:
 - **ServerName:** localhost
 - **User:** root
 - **Password:** CAADI2017
 - **DatabaseName:** CaadiIntegrada

Dar click en el botón **FINISH**.

4. Luego ahí mismo en **JDBC Connection Pools** seleccionar la Conexión que se acaba de crear, dar click y va mandar a **Edit JDBC Connection Pool** que tiene 3 pestañas: **General**, **Advanced** y **Additional Properties**. Irse a **Advanced** y deslizarse hacia abajo a **Connection Validation** y en los campos siguientes agregas los valores correspondientes:
 - **Connection Validation:** seleccionas la casilla **Required**
 - **Validation Method:** seleccionas **table**
 - **Table Name:** ahí ingresas **DUAL**

Y dar al botón **SAVE**.

IMPORTANTE Para probar que realmente se hace la conexión a la BD, te vas a **JDBC Connection Pools** seleccionar la conexión que se hizo: **MySQLConnPool** y nos mandará a **Edit JDBC Connection Pool**, dar al botón de **PING** y nos debe dar que hizo **PING** de manera exitosa. En caso de que no, revisar que hayas ingresado correctamente **ServerName**, **User**, **Password** o **DatabaseName**.

5. Ahora hay que crear el JDBC Resource. Ir a Resources > JDBC > JDBC Resources y dar al botón de NEW y asignar lo siguiente:

- **JNDI Name:** jdbc/MySQLDataSource
- **Pool Name:** Seleccionar MySQLConnPool

Y dar click al botón de OK.

6. Ahora hay que modificar en el proyecto de CAADI Migrate el archivo persistence.xml. Ahí modificar la etiqueta <jta-data-source> sustituir lo que contiene la etiqueta por jdbc/MySQLDataSource.

Y así es como se ha corregido el error del Connection Pool y un JDBC Resource.

EQUIPO 2

DESARROLLO DE LA BASE DE DATOS

La primera tarea por parte del equipo 2 fue el desarrollo de la base de datos donde se alojarían los materiales disponibles en CAADI.

Se proporcionó una base de datos que se había desarrollado en Excel por parte de otros compañeros para tomarla como base en el desarrollo de la nueva.

La base de datos fue desarrollada en MySQL tomando en cuenta algunos de los elementos de la base de datos de Excel. No todos los campos incluidos en dicha base fueron tomados en cuenta debido a la ambigüedad que algunos de esos datos causaban.

Un primer modelo entidad relación se propuso y quedó de la siguiente manera:

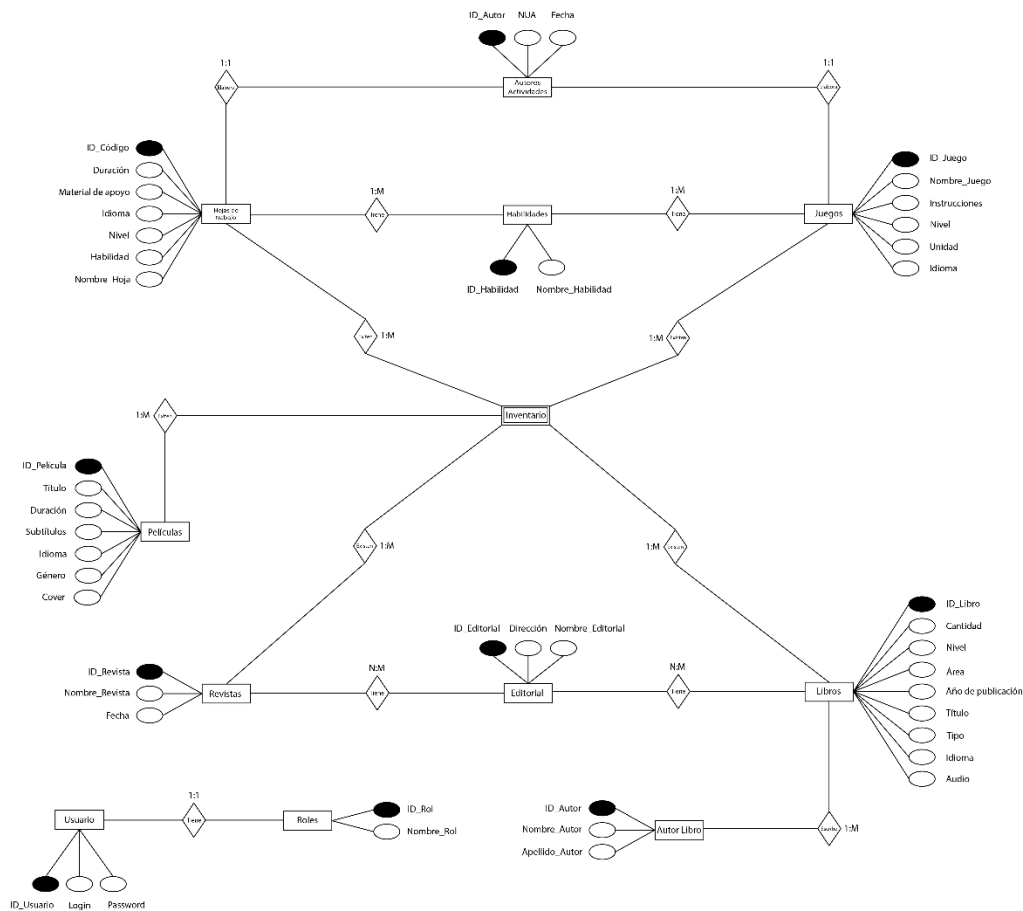


Figura 1: Primer modelo entidad-relación.

CONFLICTOS CON EL MODELO

Se presentaron demasiadas dudas con el desarrollo del modelo debido a que para el punto que nos encontrábamos aún no se tenía claro la manera en que se haría la unión de ambas bases de datos.

Los problemas fueron los siguientes:

- Asignación no apropiada para llaves primarias. Algunas entidades presentaban llaves primarias que no eran adecuadas.
- Atributos faltantes. La mayoría de las entidades no contaban con todos los atributos necesarios.
- Descomposición de entidades. Algunas entidades debían descomponerse en más de una.
- Atributos mal establecidos. Algunos atributos fueron asignados en entidades que no correspondían. Dichos atributos debían estar en las relaciones.

NUEVO MODELO DE LA BASE DE DATOS

Para poder solucionar los problemas fue necesario hacer el análisis nuevamente de la base de datos de Excel.

Con las observaciones y el análisis realizado se corrigieron los problemas y se presentó el nuevo modelo entidad-relación el cual quedó de la siguiente manera:

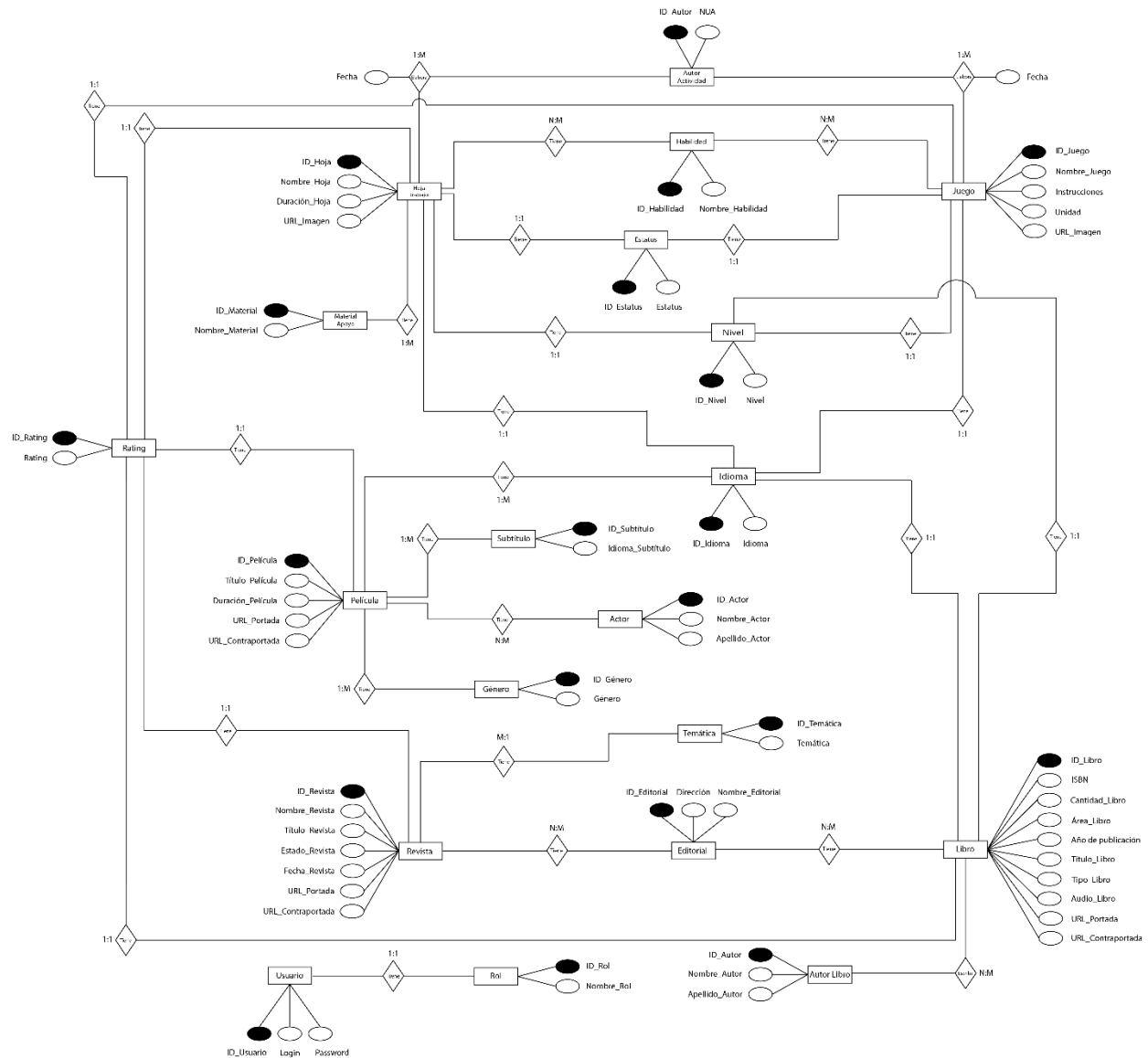


Figura 2: Segundo modelo entidad-relación propuesto.

El desarrollo del modelo fue un trabajo árduo que requirió consultas con la encargada del CAADI además de mucha dedicación y tiempo para poder lograr tener un modelo apropiado que cumpliera con todos los requerimientos establecidos.

DESCRIPCIÓN DE LOS ELEMENTOS EN LA BASE DE DATOS

Cada uno de los materiales con que cuenta CAADI deben estar presentes en la base de datos. Los materiales son los siguientes: libros, revistas, películas, hojas de trabajo y juegos. Estos dos últimos elementos son elaborados por alumnos de la carrera de LEI como parte de una materia.

No todos los materiales poseen los mismos atributos para su registro en la base de datos. A continuación se listan los atributos contemplados para cada uno de los materiales:

- Libros:
 - ID
 - ISBN
 - Cantidad
 - Área a la que pertenece
 - Año de publicación
 - Título
 - Tipo de libro
 - Editorial
 - Autor
 - Idioma
 - Rating
 - Audio (en caso de serlo)
 - URL de la portada
 - URL de la contraportada
- Revistas:
 - ID
 - Nombre
 - Título
 - Estado
 - Editorial
 - Temática
 - Rating
 - Fecha de publicación
 - URL de la portada
 - URL de la contraportada
- Películas:
 - ID
 - Título
 - Duración
 - Actor
 - Subtítulo
 - Idioma
 - Género

- URL de la portada
 - URL de la contraportada
- Hojas de trabajo:
 - ID
 - Nombre
 - Duración
 - Habilidad
 - Nivel
 - Autor
 - Idioma
 - Rating
 - Material de apoyo (en caso de que cuente con él)
 - URL (fotografía de la hoja)
- Juegos
 - ID
 - Nombre
 - Instrucciones
 - Unidad
 - Habilidad
 - Nivel
 - Autor
 - Idioma
 - Rating
 - URL (fotografía del juego)

CASOS DE USO PARA EL SISTEMA

Debido a que el sistema de CAADI puede ser usado tanto por alumnos, profesores y el administrador de CAADI, es necesario tener un caso de uso para cada uno de los usuarios.

Las tareas de cada uno de los usuarios son diferentes. Se describen a continuación la manera en que interactúan con el sistema los usuarios:

- Administrador. Es capaz de realizar cualquier tarea dentro del sistema.
- Profesores. Pueden dar de alta alumnos, generar reportes y revisar los materiales que los alumnos de LEI registran.
- Alumnos de LEI. Pueden registrar sus trabajos para su posterior evaluación.
- Alumnos. Solo pueden ingresar al sistema para registrar sus visitas a CAADI.

Los casos de uso se describen en el siguiente diagrama:

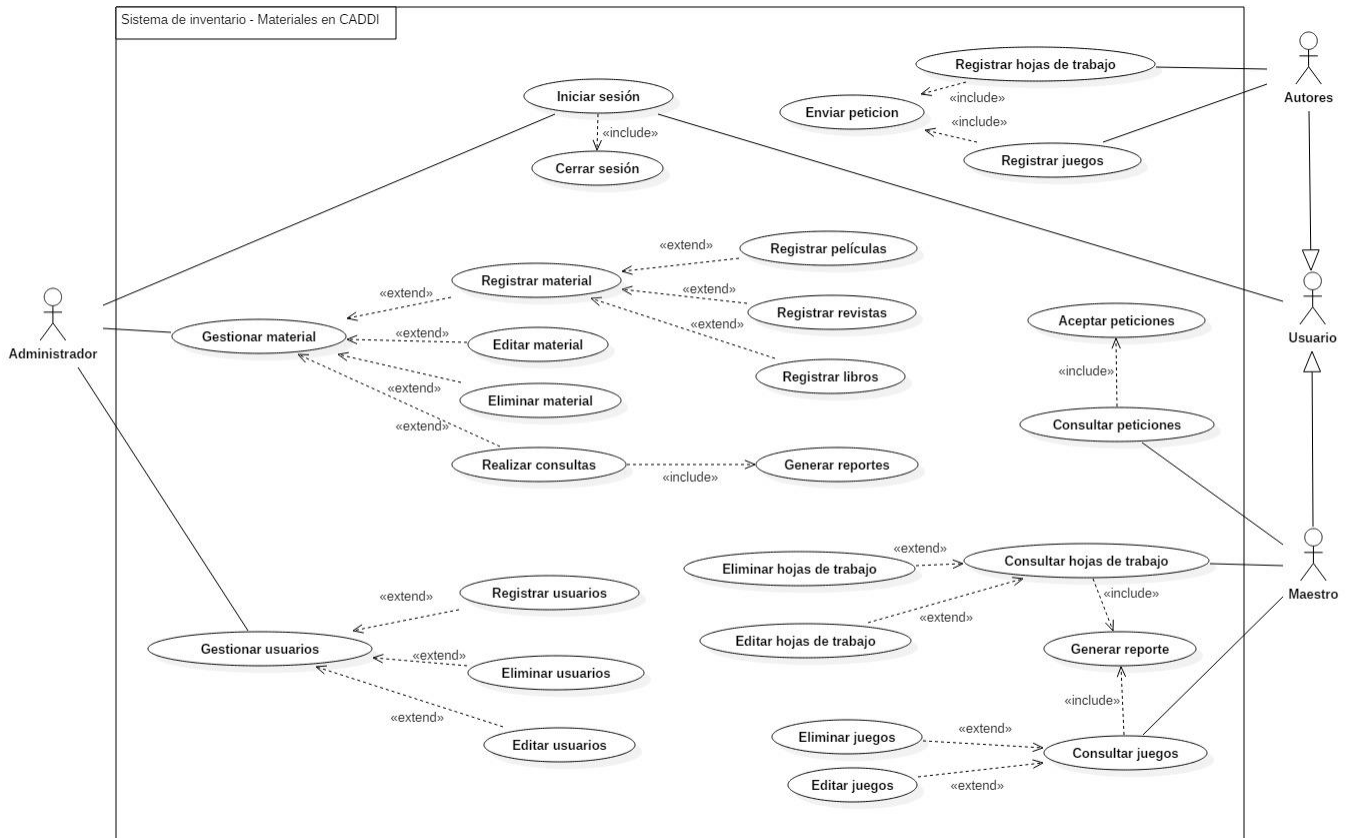


Figura 3: Diagrama de casos de uso.

MANEJO DE REPOSITORIOS

GitHub fue la plataforma en que se administró el repositorio para esta parte del proyecto. Se trabajó con un solo repositorio con las cuentas individuales de cada uno de los integrantes del equipo.

Se creó una rama para cada una de las funcionalidades que debían ser implementadas y no fue hasta que cada una de ellas tuviera una versión funcional del sistema cuando se hizo una mezcla con Develop y posteriormente con Master.

Algunas de las ramas no fueron cerradas debido a conflictos que se presentaban entre los archivos de otras versiones.

TRABAJANDO COMO UN SOLO EQUIPO

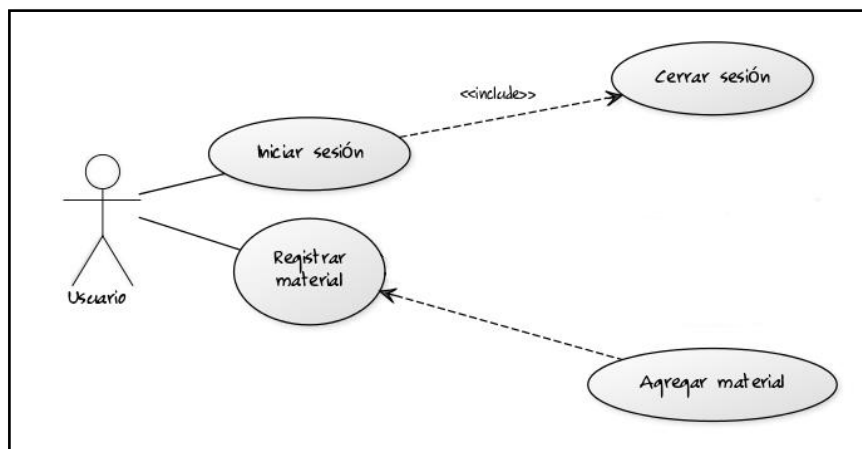
Posteriormente, el trabajo de ambos equipos de trabajo, comenzó a ser uno solo, por lo que se decidió centralizar las cuentas. Crear una en común para dar continuidad al proyecto, de manera que no dependa de usuarios en particular. El resultado fue el siguiente:

- **Username:** *proyectocaadi@gmail.com*
- **Password:** *proyectoCaadi2017*

El registro en Taiga para la asignación de las tareas fue a través de la cuenta de gmail con su respectiva contraseña. También se dispuso de una cuenta de Github para crear un repositorio del proyecto. Para dicha cuenta, se ha utilizado también la cuenta y contraseña de gmail. La liga al repositorio creado con estas credenciales, es <https://github.com/proyectoCAADI/sistemaControlDeAsistencia>.

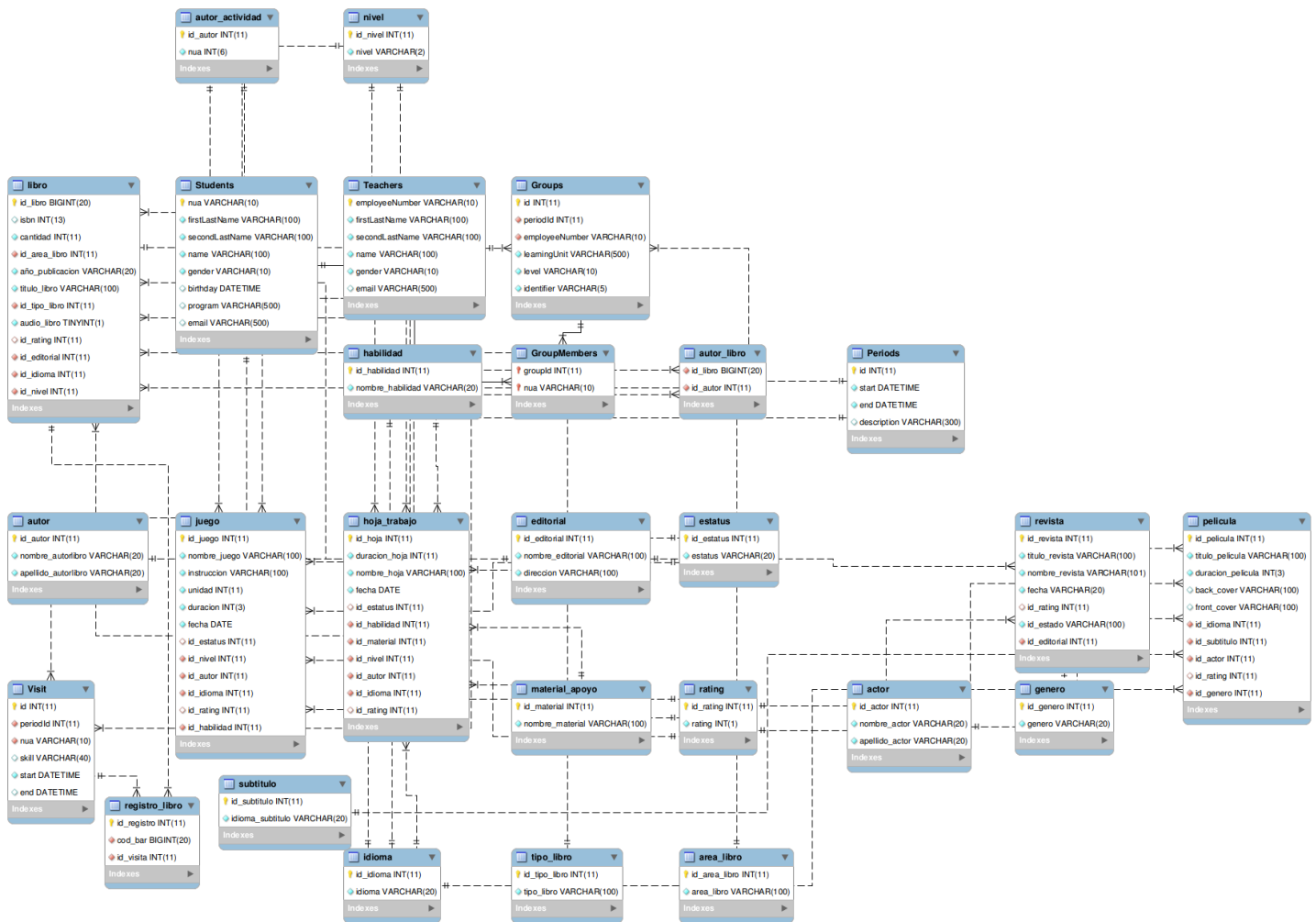
DISEÑO

El actual diseño de la interfaz principal, cuenta con la misma caja de texto y botones que cuando el equipo 1 desarrolló la interfaz, con la excepción que ahora existe un campo más en el cual se indica el número del libro que se ha utilizado. Se pensó que alumno pueda registrar el uso de material cuando finaliza sesión en CAADI. Para que un alumno inicie sesión en CAADI únicamente debe ingresar su NUA. Una vez que termine su visita, el alumno ingresa nuevamente su NUA y a continuación registra el material que uso en esa sesión. El alumno puede agregar solo un material o varios materiales en caso de que haya hecho uso de más de un material. El diseño principal no ha cambiado mucho.



MODELO RELACIONAL DE LA BASE DE DATOS ACTUAL

Tener una base de datos unificada requirió de un análisis para identificar los elementos que debían acoplarse y quitar aquellos que no fueran necesarios. Como resultado se obtuvo una base de datos de la mayoría del material disponible en CAADI.



El resto del sistema se mantuvo. Los cambios que surgieron fueron a nivel de frontend. Estos cambios se muestran en el manual de usuario-administrador.

EVALUACIÓN DE sprint 3

Los sprints anteriores se habían entregado de manera separada por parte de los dos equipos. En este sprint se han unido ambos proyectos y se trabajó en conjunto para realizar las tareas acordadas.

Para este sprint se han definido las siguientes tareas:

1. Unir las bases de datos; crear las relaciones entre las tablas que lo requieran.
2. Desarrollo de las vistas para mostrar información necesaria de la base de datos.
3. Registro de uno o más libros por parte del usuario de CAADI al finalizar sesión.
4. Realizar documentación del sistema y puesta en marcha.

Tener una base de datos unificada requirió de un análisis para identificar los elementos que debían acoplarse y quitar aquellos que no fueran necesarios. Como resultado se obtuvo una base de datos de la mayoría del material disponible en CAADI.

La finalidad del sprint fue lograr que un alumno pueda registrar el uso de material cuando finaliza sesión en CAADI. Para que un alumno inicie sesión en CAADI únicamente debe ingresar su NUA. Una vez que termine su visita, el alumno ingresa nuevamente su NUA y a continuación registra el material que uso en esa sesión. El alumno puede agregar solo un material o varios materiales en caso de que haya hecho uso de más de un material. Para este sprint, el único material disponible serán libros.

El desarrollo del sprint se llevó a cabo bajo la plataforma Java EE utilizando EJB para la parte del modelo y el controlador y JSF para la parte de la vista. Aunque no se cumplieron totalmente los puntos anteriormente mencionados, se cubrieron parcialmente. La comunicación y el trabajo en equipo salieron a relucir durante este sprint. El objetivo del último sprint fue terminar correctamente las tareas del sprint 3 y poner en marcha el sistema con las características terminadas, así como su documentación.

work hard & dream big!!!



CAADI

CAADI



MANUAL DE USUARIO

En la interfaz inicial se ingresa el NUA y se de click al botón Ingresar.

The screenshot shows a web browser window with the address bar displaying 'localhost:8080/CaadiMigrate/faces/index.xhtml'. The page features a dark blue header with the Universidad de Guanajuato logo and the text 'UNIVERSIDAD DE GUANAJUATO'. A link 'Iniciar sesión' is visible in the top right corner. The main content area is a light beige color. In the center, there is a white login form with a text input field containing '143622'. Below the input field is a blue button labeled 'Puedes empezar' with a close icon. Underneath that are two more blue buttons: 'Ingresar' and 'Salir'. At the bottom of the page, there is a light gray footer containing contact information for the Department of Multidisciplinary Studies of Yuriria.

143622

Puedes empezar

Ingresar

Salir

Department of Multidisciplinary Studies of Yuriria
Suburb Yacatitas, C.P. 38940
Yuriria, Gto., Mexico
Tel: +52 (445) 458 90 40. Exts: 1703, 1709, 1706 y 1718.
dem@ugto.mx

Después el sistema preguntará si tu NUA corresponde.

The screenshot shows the same web browser window as before, but with a modal dialog box open in the center. The dialog box has a title bar that says 'Confirmacion de usuario' and a close icon. The main text inside the dialog box asks '¿Eres correcto tu NUA?'. At the bottom right of the dialog box are two buttons: 'Aceptar' (highlighted in blue) and 'Cancelar'. The background of the page is dimmed, showing the login form from the previous screenshot.

Confirmacion de usuario

¿Eres correcto tu NUA?

Aceptar Cancelar

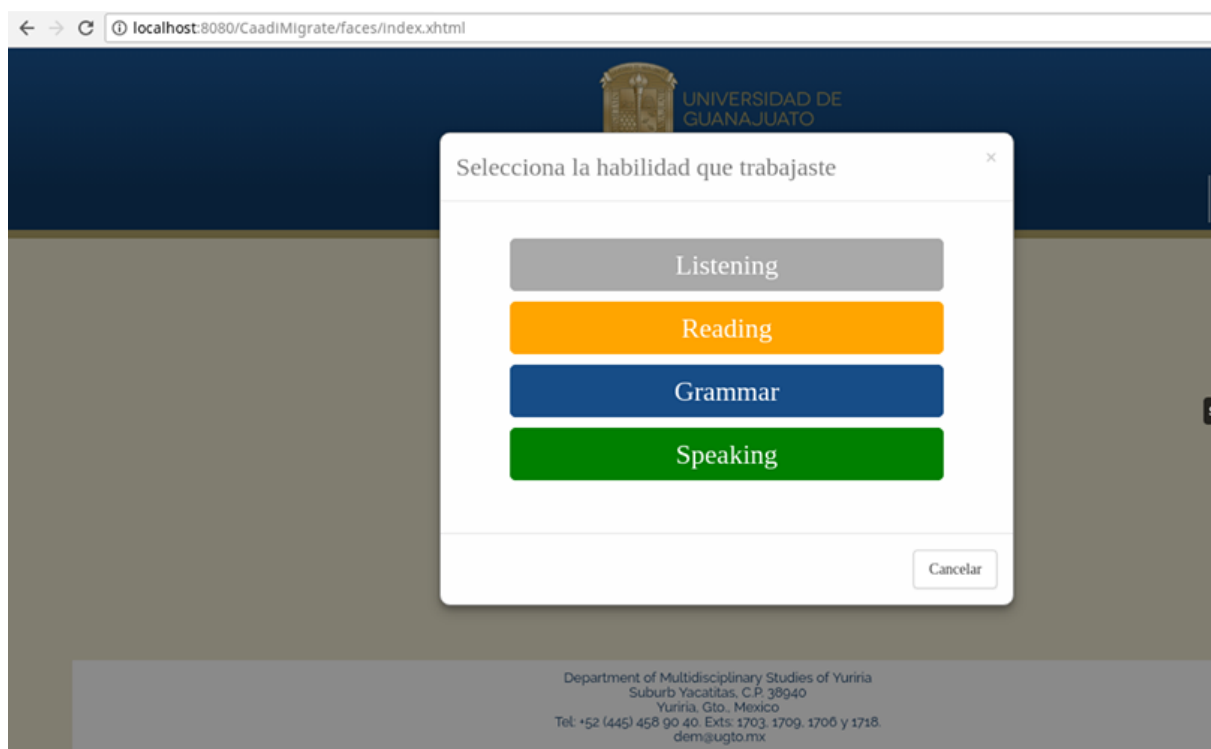
Ingresar

Salir

En caso de que sea correcto, dar clic en Aceptar y desplegará mensaje de bienvenida. En caso contrario dar click a botón cancelar e ingresar NUA nuevamente.



Se realiza la sesión y al salir se ingresa de nueva manera el NUA, sólo que ahora se mostrará el modal de actividades para poder elegir la acción realizada en el CAADI.



work hard & dream big!!!



CAADI

CAADI

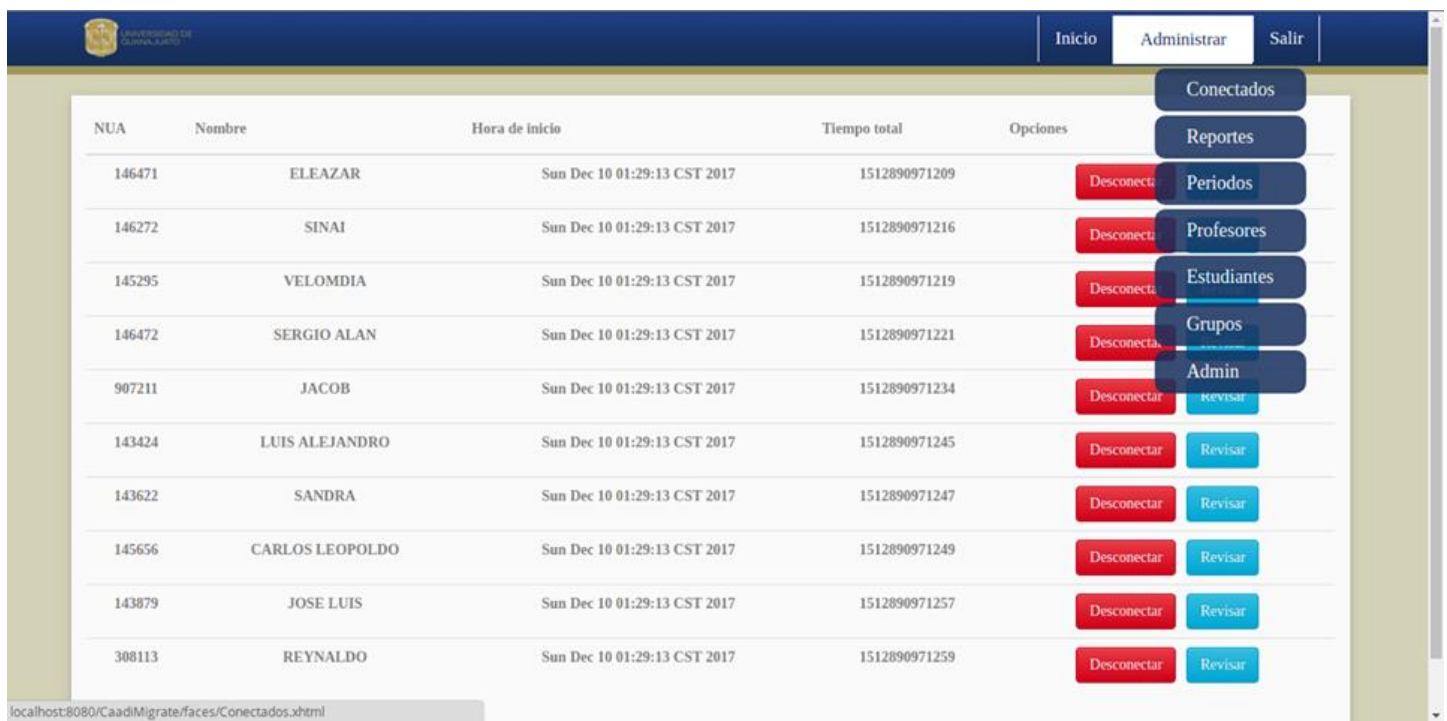


MANUAL DE ADMINISTRADOR

Respecto al administrador, este cuenta con características privilegiadas:

- Generar reportes PDF para entregar a los profesores
- Visualizar a los usuarios loggeados en el sistema
- Desconectar a los alumnos que están en el sistema pero no en el CAADI

En la pestaña Inicar Sesión, el administrador deberá ingresar **usuario:** admin **password:** admin. Una vez inició sesión se activarán 3 pestañas en la barra de navegación: *Inicio*, *Administrar* y *Salir*. La opción *Administrar* permitirá ver usuarios conectados, reportes, periodos, profesores, estudiantes, grupos.



NUA	Nombre	Hora de inicio	Tiempo total	Opciones
146471	ELEAZAR	Sun Dec 10 01:29:13 CST 2017	1512890971209	Desconectar
146272	SINAI	Sun Dec 10 01:29:13 CST 2017	1512890971216	Desconectar
145295	VELOMDIA	Sun Dec 10 01:29:13 CST 2017	1512890971219	Desconectar
146472	SERGIO ALAN	Sun Dec 10 01:29:13 CST 2017	1512890971221	Desconectar
907211	JACOB	Sun Dec 10 01:29:13 CST 2017	1512890971234	Desconectar
143424	LUIS ALEJANDRO	Sun Dec 10 01:29:13 CST 2017	1512890971245	Desconectar
143622	SANDRA	Sun Dec 10 01:29:13 CST 2017	1512890971247	Desconectar
145656	CARLOS LEOPOLDO	Sun Dec 10 01:29:13 CST 2017	1512890971249	Desconectar
143879	JOSE LUIS	Sun Dec 10 01:29:13 CST 2017	1512890971257	Desconectar
308113	REYNALDO	Sun Dec 10 01:29:13 CST 2017	1512890971259	Desconectar

En la imagen anterior se muestra la opción de los usuarios que se encuentran conectados en el sistema, permitiendo al administrador de desconectar o revisar al alumno dando click al respectivo botón.


Se muestra a continuación la interfaz de administrador para generar reportes en formato PDF con NUA, Nombre del alumno, Apellidos del alumno, el tiempo que hizo en el sistema de CAADI, el nombre del profesor y grupo.

← → ↻

localhost:8080/CaadiMigrate/faces/reportes.xhtml

☆

⋮

UNIVERSIDAD DE GUAYAQUIL

Inicio | Administrar | Salir

Nombre del profesor

Grupo

Opciones

JANETTE LOPEZ MIRANDA ▾

learningUnit 3 ▾

Listar alumnos

Crear PDF

NUA	Nombre	Apellidos	Tiempo
143879	JOSE LUIS	CORONA MELENDEZ	0:0:0
145295	VELOMDIA	DARIO JIMENEZ	0:0:0
145656	CARLOS LEOPOLDO	ROJAS DELGADO	0:0:0
145867	RODRIGO	GUZMAN MALDONADO	0:0:13
146068	HUGO ARMANDO	AGUILERA GARCIA	0:0:0
335912	HECTOR GANDHI	FLORES SALDIVAR	0:0:0
907201	URIEL	CALDERON URIBE	0:0:0
907211	JACOB	GONZALEZ VILLAGOMEZ	0:0:0