**SOLUCIÓN GAMIFICACIÓN MATEMÁTICA ESTRUCTURAL Y LÓGICA**

En este documento se describe la solución implementada para el curso de Matemática Estructural y Lógica de la facultad de Ingeniería. El propósito de la solución es facilitar la aplicación de una estrategia pedagógica de gamificación, que permita a los estudiantes registrar su desempeño en las diferentes actividades y a los profesores gestionar las diferentes etapas del curso en el transcurso del periodo académico.

1. **Descripción general del dashboard**

Como parte de la solución se desarrolló un dashboard sobre Grails, el cual permite a los estudiantes visualizar su desempeño individual y grupal tanto de manera semanal como general, y a los profesores visualizar el estado de cada uno de los estudiantes y equipos que conforman sus secciones a cargo. A continuación, se muestra la vista principal del dashboard con la cual interactúan tanto los estudiantes como los profesores:



Figura 1: vista principal del dashboard

La vista principal tiene dos pestañas, una para visualizar los datos por equipos y la otra para visualizar los datos de un estudiante específico. En el caso del estudiante, este sólo puede consultar su información individual y aquella de los equipos que componen su sección. En el caso del profesor, este puede seleccionar un estudiante para consultar su información individual y aquella de los equipos que componen la sección a la que pertenece.

En la pestaña por equipos se muestra para cada equipo:

* El nombre.
* Los logins de los estudiantes que lo componen, ordenados ascendentemente por puntos.
* Los puntos de cada miembro con la sumatoria y el promedio general.
* Las monedas de cada miembro con la sumatoria general y la cantidad disponible para compras.
* Las medallas de cada miembro.

En la pestaña individual (Figura 2) se muestra para el estudiante:

* Cantidad de monedas ganadas por semana.
* Porcentaje de aporte semanal en monedas al equipo.
* Número de gemas ganadas.
* Número de puntos ganados.
* Número de medallas ganadas.
* Número de monedas ganadas en total.



Figura 2: pestaña individual del dashboard

1. **Descripción detallada del dashboard**

En la Figura 3 se muestra la vista de funciones disponibles para los usuarios administradores (profesores y monitores) y super administradores.

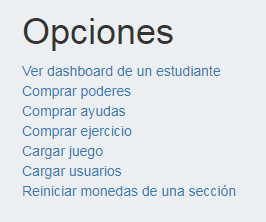


Figura 3: funciones de administrador

La primera opción (Ver dashboard de un estudiante) lleva al usuario a la vista principal descrita anteriormente. Esta función está disponible para los administradores, quienes pueden ver la información de los estudiantes de sus secciones a cargo, y para los super administradores, quienes pueden ver la información de los estudiantes de todas las secciones del curso.

A continuación, se describen las demás funcionalidades disponibles.

* 1. **Compra de poderes**

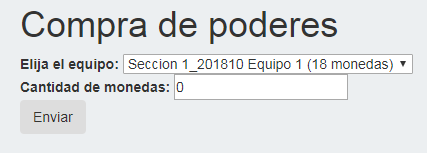


Figura 4: vista para comprar poderes

Por medio de esta función, los administradores y super administradores pueden simular la compra de un poder dentro de la dinámica del curso. Esta compra la paga el equipo con las monedas totales acumuladas por sus miembros. El usuario ingresa el número de monedas que cuesta el poder, y estas son descontadas inmediatamente al hacer clic sobre el botón Enviar. En caso que el equipo no tenga las monedas requeridas para la compra, se muestra un mensaje de error indicándolo y no se afecta la cantidad de monedas del mismo.

* 1. **Compra de ayudas**

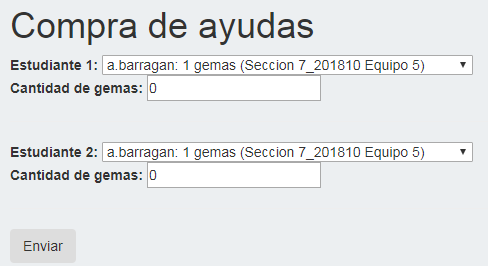


Figura 5: vista para comprar ayudas

Por medio de esta función, los administradores y super administradores pueden simular la compra de ayudas dentro de la dinámica del curso. Esta compra la paga una pareja de estudiantes o un único estudiante. El usuario ingresa el número de gemas que paga cada estudiante por la ayuda, y estas son descontadas inmediatamente al hacer clic sobre el botón Enviar. En caso que alguno de los estudiantes no tenga las gemas requeridas para la compra, se muestra un mensaje de error indicándolo y no se afecta la cantidad de gemas de los mismos. Para realizar la compra con un único estudiante sólo se ingresan los datos para el estudiante 1 (automáticamente se llenan los campos del estudiante 2 con los datos del estudiante 1, pero dejando la cantidad de gemas en cero).

* 1. **Compra de ejercicio**

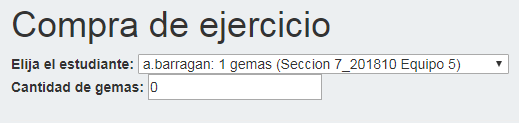


Figura 6: vista para comprar un ejercicio

Por medio de esta función, los administradores y super administradores pueden simular la compra de un ejercicio dentro de la dinámica del curso. Esta compra la paga un estudiante. El usuario ingresa el número de gemas que paga el estudiante por el ejercicio, y estas son descontadas inmediatamente al hacer clic sobre el botón Enviar. En caso que el estudiante no tenga las gemas requeridas para la compra, se muestra un mensaje de error indicándolo y no se afecta la cantidad de gemas del mismo.

* 1. **Reinicio de monedas**

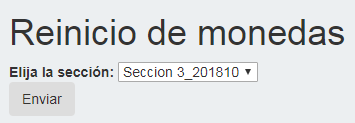


Figura 7: vista para reiniciar las monedas de una sección

Por medio de esta función, los administradores y super administradores pueden reiniciar en ceros las monedas de todos los equipos de una sección del curso. El reinicio se efectúa inmediatamente al hacer clic sobre el botón Enviar. Después de reinicio los miembros de los equipos deben completar más actividades para acumular así monedas disponibles para compras.

* 1. **Cargar usuarios**

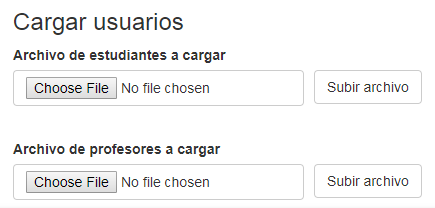


Figura 8: vista para cargar estudiantes y administradores

Por medio de esta función los super administradores pueden cargar, desde archivos en formato csv, el conjunto de estudiantes de todas las secciones de un semestre y los profesores con los monitores a cargo de estas. A continuación, se muestra una captura del formato csv para la carga de administradores.

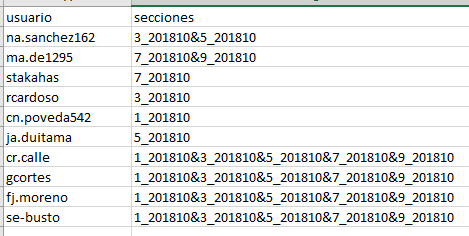


Figura 9: formato csv para cargar profesores y monitores

En la primera columna se indica el login del usuario. En la segunda columna se indican los nombres de las secciones a cargo del usuario, separadas por el caracter &. Los nombres de las secciones incluyen el periodo académico para diferenciarlas dentro del motor de gamificación que subyace en la infraestructura.

Cada vez que se carga el archivo de administradores (profesores y monitores), se crea el usuario en la base de datos (si no existe) y se asignan las secciones de acuerdo con el archivo. Si el usuario tenía otras secciones asignadas, estas serán borradas y el usuario quedará sólo con aquellas indicadas en el csv. En el archivo se incluyen los login de los super administradores junto con todas las secciones del curso en el semestre actual, para permitirles consultar la información de cualquier estudiante y equipo en caso de ser necesario. Las secciones que se indiquen para cada usuario en este archivo serán las únicas que el usuario podrá consultar desde el dashboard.

A continuación, se muestra una captura del formato csv para la carga de estudiantes.



Figura 10: formato csv para cargar estudiantes

En la primera columna se indica el login del estudiante. En la segunda columna se indica el nombre de la sección a la que pertenece el estudiante. En la tercera columna se indica el equipo, dentro de la sección, al que pertenece el estudiante.

Cada vez que se carga el archivo de estudiantes se crea el usuario en la base de datos (si no existe) y se conforman los equipos dentro del motor de gamificación, dejando todos los valores (monedas, gemas, puntos, misiones completadas, etc.) en ceros. Este cargue sólo es necesario al inicio del semestre, pero puede efectuarse en cualquier momento para actualizar los miembros de los equipos o para volver a completar misiones con notas corregidas o diferentes. Aunque este cargue reinicia el juego, se puede recuperar el estado nuevamente al cargar el último archivo de notas de Sicua. Sólo se pierden las cantidades de monedas disponibles para compras de los equipos, las cuales se pueden reasignar simulando compras después del reinicio. Si un estudiante ya existe en la base de datos y en el motor de gamificación al momento de efectuarse el cargue, porque tomó el curso en un semestre previo, se actualizan su sección y equipo al semestre actual perdiendo todos los datos del semestre previo.

* 1. **Cargar juego**

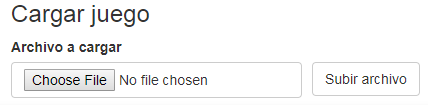


Figura 11: vista para cargar juego de Sicua

Por medio de esta función los super administradores pueden cargar las notas provenientes de Sicua, simulando así la dinámica del juego. Aunque el archivo de carga tiene múltiples columnas, sólo se envían al motor de gamificación los campos correspondientes al nombre de la actividad y al porcentaje obtenido en la actividad por cada estudiante. Este archivo es generado automáticamente por un proceso administrado por la DSIT, y aunque se genera diariamente, siempre el archivo más reciente contiene la información de los archivos más antiguos. Lo anterior implica que sólo con cargar el archivo más reciente, se recupera el estado del juego completamente desde su fecha de inicio hasta la fecha de generación de dicho archivo.

No obstante el cargue puede efectuarse manualmente a través de esta funcionalidad, actualmente está automatizado mediante una tarea diaria en el crontab del usuario root dentro del servidor *union*. A continuación, se muestra una captura del script */datos/cargueMEL/cargueMEL,* ejecutado por dicha tarea.

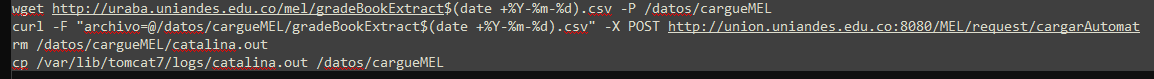


Figura 12: script cargueMEL

Mediante este script se:

* Obtiene una copia del archivo de notas de Sicua a cargar.
* Cargan y pasan los datos al motor de gamificación para simular la dinámica del juego.
* Obtiene una copia actualizada del log del servidor para incluir la traza con los resultados del cargue.

Las copias de los archivos de notas y del log del servidor se guardan en la ruta */datos/cargueMEL/*

Dado que el dashboard utiliza los servicios de la capa que brinda el motor de gamificación, no hay control directo sobre los puntos, gemas, monedas y medallas que se otorgan de acuerdo con el porcentaje obtenido en una actividad. Así mismo, el motor interpreta cada actividad como una misión en el mundo gamificado, la cual permite sólo un intento y en consecuencia un único porcentaje de desempeño que después no puede modificarse. Esto hace inviable cambiar el porcentaje de un estudiante para una actividad que ya fue calificada. Por tanto, si se quiere asignar otra nota a una o varias actividades ya calificadas, se debe seguir el siguiente procedimiento:

1. Modificar las notas de los estudiantes en Sicua.
2. Cargar nuevamente el archivo csv de estudiantes en el dashboard para reiniciar el juego.
3. Esperar el cargue automático del archivo de notas de Sicua para recuperar el estado del juego hasta la fecha, pero incluyendo esta vez los cambios hechos sobre las notas.
4. Dado que en el punto anterior no se recuperan las cantidades de monedas disponibles para compras de los equipos, estas deben ser reasignadas manualmente simulando compras de poderes en el dashboard, de ser necesario.
5. **Motor de gamificación**

Como parte de la infraestructura, se cuenta con un motor de gamificación que permite simular el juego a partir de las actividades y porcentajes de desempeño que se toman de Sicua. Para integrarlo a la solución, fue necesario definir una capa de servicios del motor y una capa de servicios de la aplicación que lo utiliza.

Los servicios que ofrece la capa de la aplicación *(AppService)* son:

* traerDatosEstudiante
* traerEquipo
* traerSeccion
* crearEstudiante
* eliminarEstudiante
* crearEquipo
* eliminarEquipo
* registrarPrueba
* gastarGemasEstudiante
* gastarGemasGrupo
* gastarMonedasEquipo
* reiniciarMonedasEquipo
* reiniciarMonedasSeccion

Los servicios que ofrece la capa del motor *(MotorService)* son:

* getPlayerData(String app, String idPlayer)
* getTeamData(String app, String idTeam)
* getTeams(String app)
* getMarkets(String app)
* getMarketForPlayer(String app, String idMarket, String idPlayer)
* createPlayer(String app, String idPlayer, String name, String email, String idTeam)
* joinTeam(String app, String idTeam, String idPlayer)
* acceptMission(String app, String idMission, String idPlayer)
* rejectMission(String app, String idMission, String idPlayer)
* completeMission(String app, String idMission, String idPlayer, String scores, String rewards)
* buyItemForPlayer(String app, String idPayment, String idPlayer)
* buyItemForTeam(String app, String idPayment, String idTeam)
* spendPlayerCurrencies(String app, String idCurrency, String quantity, String idPlayer)
* spendTeamCurrencies(String app, String idCurrency, String quantity, String idTeam)
* createTeam(String app, String name, String idTeam)
* removeCurrenciesFromPlayer(String app, String idCurrency, String quantity, String idPlayer)
* removeCurrenciesFromTeam(String app, String idCurrency, String quantity, String idTeam)
* changeTeam(String app, String idTeam, String idPlayer)
* deletePlayer(String app, String idPlayer)
* deleteTeam(String app, String idTeam)

