**Historias de Usuario para la aplicación**

**CalcLogic**

* 1. Estructura de las Historias de Usuario:

1) "En tanto [Usuario] hace [acción] para conseguir [beneficio].

2) **ID:** "nombre del requerimiento"

**Como:** [rol]

**Quiero:** [acción en el sistema]

**Para:** [fin especifico con el sistema].

Criterios de Aceptación:

\*) Condición que se quiere, características, recursos.

* 1. Historias de Usuario
     1. Login de la aplicación:

**ID:** "hacer un login para la aplicación" – LOG-1

**Como:** Usuario de la aplicación

**Quiero:** Autenticarme como usuario registrado

**Para:** Entrar a la aplicación

**Criterio de Aceptación:**

\*) Seguridad.

\*) Caracteres de contraseña en "\*".

\*) Debe ser capaz de recordar contraseña.

* + 1. Guardar teorema:

1. **ID:** " Guardar teorema propio" – GUAR/TEO-1.
2. **Como:** Usuario
3. **Quiero:** Guardar un teorema
4. **Para:** Luego realizar una demostración del enunciado.
6. **Criterio de Aceptación:**

* El formulario para ingresar teorema debe tener los campos enunciado, teoría, categoría, número del teorema y nombre de teorema, además de una lista de abreviaciones preestablecidas y propias disponibles para usarlas en el enunciado.
* No se debe dejar el campo del enunciado del teorema vacío.
* No se debe dejar el campo del número del teorema vacío
* El número de teorema no puede ser repetido
* El enunciado del teorema no puede ser alfa-equivalente a algún otro teorema.
* Todos los errores del parse al leer el enunciado deben imprimirse como mensajes de error
* Un error en el formulario debe volver a la misma vista recordando lo que estaba escrito en los formularios (no se debe perder información en los formularios luego de un error)
* Los errores de que un simbolo o abreviación en el enunciado no se encuentra en la base de datos, debe ser indicado al usuario.
* Al guardar el teorema, se lista como un teorema no demostrado aún.

1. **ID:** " Guardar teorema predeterminado" – GUAR/TEO-2.
2. **Como:** AdminTeoremas
3. **Quiero:** Guardar un teorema
4. **Para:** Que cualquier usuario pueda realizar una demostración del enunciado.
6. **Criterio de Aceptación:**

* El formulario para ingresar teorema debe tener los campos enunciado, teoría, categoría, número del teorema y nombre de teorema, la lista de abreviaciones preestablecidas y propias disponibles para usarlas en el enunciado, y además un campo para indicar si el teorema es axioma o no.
* No se debe dejar el campo del enunciado del teorema vacío.
* No se debe dejar el campo del número del teorema vacío
* Todos los errores del parse al leer el enunciado deben imprimirse como mensajes de error
* Un error en el formulario debe volver a la misma vista recordando lo que estaba escrito en los formularios (no se debe perder información en los formularios luego de un error)
* Los errores de que un simbolo o abreviación en el enunciado no se encuentra en la base de datos, debe ser indicado al usuario.
* Al guardar el teorema, se lista como un teorema no demostrado aún para todos los usuarios al menos que se seleccione que el enunciado es un axioma, en este caso será un axioma para todos los usuarios.
* Al agregar un teorema, debe ser visible inmediatamente a todos los usuarios que estén registrados y a todos aquellos usuarios luego del momento del registro.
  + 1. Listar teoremas:

1. **ID:** " Listar teoremas" – LIST/TEO.
2. **Como:** Usuario
3. **Quiero:** Listas mis teoremas
4. **Para:** Luego realizar una demostración del enunciado.
6. **Criterio de Aceptación:**

* Se listan los enunciados de todos los axiomas y teoremas demostrados o no, con sus respectivos nombres y número de identificación.
* Los axiomas y teoremas con por lo menos una demostración completa vienen acompañados de un candado abierto. El resto de los teoremas vienen acompañados de un candado cerrado.
* Los teoremas con por lo menos una demostración completa son clickeables y al hacer click aparece la lista de las demostraciones de ese teorema. Al hacer click sobre algún elemento de esta lista hace aparece la demostración asociada al item.
* Si la demostración del teorema es dada de forma automática, por medio de un algoritmo que demuestra un metateorema, igualmente se considera como un teorema clickeable y al hacer click aparece la lista con las demostraciones tal que al hacer click en un item se pueden ver las demostraciones generadas.
* Algunos teoremas se encuentran juntos bajo un mismo número identificador y nombre. Dichos teoremas aparecen acompañados de un símbolo +. Al hacer click sobre el símbolo + se despliegan todos los teoremas asociados al número-nombre con identificadores independientes. (no tenemos forma de configurar qué teoremas estan asociados a qué número-nombre, por ahora lo hacemos manual desde la base de datos).
  + 1. G
    2. Metateoremas:

1. **ID:** " Demostrar por medio de algoritmos de prueba" – DEM/ALG.
2. **Como:** Usuario
3. **Quiero:** Generar automáticamente demostraciones de teoremas
4. **Para:** No tener que hacer manualmente pasos de inferencias repetitivos o predecibles.
6. **Criterios de Aceptación:**

* Un algoritmo de prueba usa algunos teoremas para realizar las inferencias. Por lo tanto el algoritmo tiene como precondición, que la totalidad de los teoremas que usa el algoritmo, estén ya demostrados por el usuario. Al cumplirse la precondición, los teoremas que no tengan una demostración pero se pueda generar automáticamente por el algoritmo, no deben ser clickeables en la lista de mis teoremas.
* Al demostrar los teoremas que forman la precondición del algoritmo, inmediatamente, todos los teoremas que no tienen demostración, pero son generables por el algoritmo, se vuelven clickeables en la lista de mis teoremas, para así poder visualizar las demostraciones generadas.
  + 1. Uso de abreviaciones:

I**D:** " Uso de Abreviaciones preestablecidas en las formulas de la aplicación" – USO 1.

**Como:** Usuario

**Quiero:** Usar varias abreviaciones preestablecidas

**Para:** formar una fórmula en la aplicación

**Criterio de Aceptación:**

* Deben estar disponibles en una lista <a seleccionar>, todos las Abreviaciones predeterminadas, en toda vista donde exista un campo de un formulario, donde se deba ingresar una fórmula.
* En todo campo donde se ingrese una fómula, es posible usar la abreviación, escribiendo como un subtérmino de la fórmula, la abreviación con sus argumentos en notación funcional Alias(p,q,..,r), donde Alias es el nombre de la abreviación en la BD
* Si al escribir una fórmula, se usa una abreviación que no se encuentra en la BD, se debe mostrar un mensaje de error que indique que la abreviación no existe
  + 1. Introducción de abreviaciones:

**ID:** " Introducción de Abreviaciones propias en las lista de la aplicación"– INTRO-1.

**Como:** Usuario

**Quiero:** Introducir Abreviación propia

**Para:** ser guardada en la base de datos

**Criterio de Aceptación:**

* Debe verificar que el alias no está repetido entre las abreviaciones propias o predefinidas.
* No se debe dejar el campo del Alias vacío.
* No se debe dejar el campo de fórmula vacío.
* Debe verificar que la sintaxis este correcta, mostrando todos los mensajes de error del parse.
* Todos los Alias deben empezar por mayúscula y seguir con minúsculas. Los parámetros del Alias deben ser solo variables y estar entre paréntesis separadas por coma. Si no hay parámetros se debe poner al final del Alias un parentesis que abre y cierra sin nada en el medio.
* Debe verificar que no es alpha equivalente a una abreviación propia o predefinida que ya existe, mostrando un mensaje de error.
* Debe verificar que las variables que se usan como parámetros de la abreviación, son las mismas que las variables que usa la fórmula que se está abreviando.
* Luego de introducir la abreviación, debe poderse buscar la abreviación en la lista de abreviaciones propias, usando como parámetros las mismas variables (y en el mismo orden) que se usaron al guardar la abreviación.
* No se debe permitir una abreviación con un solo argumento, en donde la fórmula a abreviar es exactamente ese argumento.
* Abreviación propia de un usuario no debe verlo otro usuario.
* Si ya existe el alias o una abreviación alpha equivalente en otro usuario, debe poderse introducir igualmente.
* Debe existir un campo de notación de la abreviación para imprimir la notación en latex. Por ejemplo una notación válida es \lim\_{%(a1)}%(a2)
* Depués de un mensaje de error, se debe poder visualizar el contenido de un formulario antes del mensaje de error. (No perder información de formularios).
* Si algún subtérmino de la fórmula a introducir es un alias que no existe en la lista de abreviaciones predefinidas y propias, se debe generar un mensaje de error que notifique que la abreviación no existe.

**ID:** " Introducción de Lambda-términos preestablecidos en las listas de la aplicación "–

INTRO-2.

**Como:** AdminTeoremas

**Quiero:** introducir Abreviación preestablecidos en las listas de la aplicación

**Para:** ser guardados en la base de datos.

**Criterio de Aceptación:**

* Debe verificar que el alias no esté repetido entre las abreviaciones predeterminandas.
* No se debe dejar el campo del Alias vacío.
* No se debe dejar el campo de fórmula vacío.
* Debe verificar que la sintaxis este correcta, mostrando todos los mensajes de error del parse.
* Todos los Alias deben empezar por mayúscula y seguir con minúsculas. Los parámetros del Alias deben ser solo variables y estar entre paréntesis separadas por coma. Si no hay parámetros se debe poner al final del Alias un parentesis que abre y cierra sin nada en el medio.
* Debe verificar que no es alpha equivalente a una abreviación predefinida que ya existe.
* Debe verificar que las variables que se usan como parámetros de la abreviación son las mismas que las variables que usa la fórmula que se está abreviando.
* Luego de introducir la abreviación, debe poderse buscar la abreviación en la lista de abreviaciones predefinidas, usando como parámetros las mismas variables (y en el mismo orden) que se usaron al guardar la abreviación.
* Abreviación predefinida debe poder verlo todos los usuario.
* Debe existir un campo de notación de la abreviación para imprimir la abreviación en latex. Por ejemplo una notación válida es \lim\_{%(a1)}%(a2)
* Depués de un mensaje de error, se debe poder visualizar el contenido de un formulario antes del mensaje de error. (No perder información de formularios).
* Si algún subtérmino de la fórmula a introducir es un alias que no existe en la lista de abreviaciones predefinidas y propias, se debe generar un mensaje de error que notifique que la abreviación no existe.
* Toda abreviación preestablecida, es una abreviación propia para el usuario AdminTeoremas
  + 1. Modificación de abreviaciones:

I**D:** " Modificación de Abreviación propia en la aplicación" – MOD-1.

**Como:** Usuario

**Quiero:** Modificar Abreviación propias

**Para:** ser almacenado para su uso posterior

**Criterio de Aceptación:**

* Luego de modificar la abreviación, cuando se busca en las listas de abreviaciones propias, se obtiene su versión actualizada, usando las variables (y en el mismo orden) de los parámetros que usaron en la modificación. Se puede buscar el término nuevo y no debe aparecer el viejo
* Cuando se elige modificar una abreviación, debe aparecer en la siguiente vista, la abreviación a modificar, el alias (con los parámetros) y notación, escritas en los campos de texto, tal cual como se encuentre en base de datos en ese momento.
* El enlace a la vista de modificación, debe encontrarse en la fila donde se encuentra la abreviación a modificar en la tabla de abreviaciones propias.
* Al enviar el formulario de la vista de modificación, se debe suprimir la verificación de si es igual el alias o la fórmula es alfa-equivalente a la versión previa de la abreviación
* Debe cumplir todos los criterios de aceptación de INTRO-1.
* Si una abreviación es modificada y anteriormente fue usada en una fórmula, entonces la versión que se encuentra en la fórmula no cambia al menos que se cambie solo el campo de notación.

**ID:** "Modificar lambda-términos preestablecidos de la aplicación" – MOD-2

**Como:** AdminTeoremas

**Quiero:** Modificar Abreviaciones preestablecidos

**Para:** que estos estén disponibles para todos los ususarios.

**Criterio de Aceptación:**

* Luego de modificar la abreviación, cuando se busca en las listas de abreviaciones predeterminadas, se obtiene su versión actualizada, usando las variables (y en el mismo orden) de los parámetros que usaron en la modificación. Se puede buscar el termino nuevo y no debe aparecer el viejo
* Cuando se elige modificar una abreviación, debe aparecer en la siguiente vista, la abreviación a modificar, el alias (con los parámetros) y notación, escritas en los campos de texto, tal cual como se encuentre en base de datos en ese momento.
* El enlace a la vista de modificación, debe encontrarse en la fila donde se encuentra la abreviación a modificar en la tabla de abreviaciones predeterminadas, desde la sesión de AdminTeoremas.
* Al enviar el formulario de la vista de modificación, se debe suprimir la verificación de si es igual el alias o la fórmula es alfa-equivalente a la versión previa de la abreviación.
* Debe cumplir todos los criterios de aceptación de INTRO-2.
* Si una abreviación es modificada y anteriormente fue usada en una fórmula, entonces la versión que se encuentra en la fórmula no cambia al menos que se cambie solo el campo de notación.
  + 1. Eliminación de abreviaciones

**ID:** "Eliminar abreviaciones en una cuenta activa determinada de la aplicación" – ELIM-1

**Como:** Usuario

**Quiero:** Borrar abreviaciones propias asociadas a mi cuenta

**Para:** que estos ya no estén en la lista de abreviaciones propias de la cuenta determinada.

**Criterio de Aceptación:**

* Por ser eliminaciones debe pedir confirmación de la acción.
* El boton de eliminar se debe encontrar en la fila de la abreviación a borrar de la tabla de abreviaciones propias.
* Luego de eliminar la abreviación debe desaparecer de la lista de abreviaciones propias
* Si una abreviación es eliminada, no puede ser usada para construir una fórmula. Sin embargo, si esta abreviación ha sido usada en alguna fórmula, la fórmula mantiene la abreviación pero en notación funcional con la lista de parámetros separados por coma y entre paréntesis.

**ID:** " Eliminación de abreviaciones preestablecidas en las listas de la aplicación "–

ELIM-2.

**Como:** AdminTeoremas

**Quiero:** eliminar abreviación preestablecida en las listas de la aplicación

**Para:** ser retirada de la base de datos.

**Criterio de Aceptación:**

* Por ser eliminaciones debe pedir confirmación de la acción.
* El boton de eliminar se debe encontrar en la fila de la abreviación a borrar de la tabla de abreviaciones propias de la sesión de AdminTeoremas.
* Luego de eliminar la abreviación debe desaparecer de la lista de abreviaciones predeterminadas para todos los usuarios.
* Si una abreviación es eliminada, no puede ser usada por ingún usuario para construir una fórmula. Sin embargo, si esta abreviación ha sido usada en alguna fórmula por algún usuario, la fórmula mantiene la abreviación pero en notación funcional con la lista de parámetros separados por coma y entre paréntesis.
  + 1. Listar abreviaciones

**ID:** "Listar abreviaciones propias" -LIST-1

**Como:** Usuario

**Quiero:** Listar abreviaciones propias.

**Para:** poder seleccionar la acción eliminar o modificar.

**Criterio de Aceptación:**

* Debe aparecer el alias de cada abreviación acompañado de los parámetros con las variables que se usaron cuando se guardó en base de datos.
* Por cada abreviación debe aparecer el campo con la fórmula que se está abreviando
* Por cada abreviación debe aparecer el campo de notación
* Por cada abreviación deben aparecer las acciones: modificar y eliminar.

**ID:** "Listar abreviaciones predeterminadas" -LIST-2

**Como:** AdminTeoremas

**Quiero:** Listar abreviaciones predeterminadas.

**Para:** poder seleccionar la acción eliminar o modificar.

**Criterio de Aceptación:**

* Debe aparecer el alias de cada abreviación acompañado de los parámetros con las variables que se usaron cuando se guardó en base de datos.
* Por cada abreviación debe aparecer el campo con la fórmula que se está abreviando
* Por cada abreviación debe aparecer el campo de notación
* Por cada abreviación deben aparecer las acciones: modificar y eliminar.

* + 1. Inspeccionar abreviaciones

**ID:** "Inspeccionar abreviaciones dentro de una fórmula" - INSP-1

**Como:** Usuario

**Quiero:** Poder examinar las abreviaciones que estan como subtérminos de una fórmula.

**Para:** Observar la abreviación representada en términos de operadores básicos.

**Criterios de Aceptación:**

* Las abreviaciones tienen parámetros, estos parámetros pueden ser fórmulas y pueden contener abreviaciones, deben podeser ver los parámetros según como lo indique el campo de notación de la abreviación. Las abreviaciones se reconocen en una fórmula porque se imprimen de color azul, salvo los parámetros que son de color negro como el resto de la fórmula (salvo las abreviaciones que contengan los parámetros que también son de color azul).
* Se deben poder visualizar la fórmula que se está abreviando haciendo click sobre la abreviación.
* Si la fórmula que se está abreviando tiene abreviaciones, entonces la fórmula que se visualiza al hacer un click se muestra en términos de las subabreviaciones. Se debe hacer un segundo click sobre estas subabreviaciones para poder visualizarlas, y así sucesivamente.
* Una vez que haciendo click se visualiza una fórmula F que estaba oculta por la abreviación, se puede volver a ocultar esta fórmula haciendo click nuevamente encima de F, menos en las zonas de F que existan subabreviaciones. Si se habían descubierto las fórmulas de subabreviaciones que contenía F, entonces se deben ocultar primero las fórmulas de las subabreviaciones, antes de poder hacer click para ocultar la fórmula que contiene las subabreviaciones
* Cuando se visualiza una fórmula F que estaba oculta por una abreviación, debe aparecer F subrayada con una línea azul, el espacio que ocupa esta línea azul, se considera como un sector clickeable como para volver a ocultar F y mostrar la abreviación.
* Hacer click sobre los parámetros no muestra la fórmula oculta, al menos que se esté haciendo click sobre una abreviación que contiene el parámetro, en ese caso se muestra la fórmula oculta de esa abreviación del parámetro solamente.
* A una abreviación que se encuentra en una fórmula se le puede cambiar su presentación al modificar el campo de notación en el módulo de modificar abreviación.
  + 1. Ocultar y mostrar categorías

**ID:** "Seleccionar categorías a mostrar en la vista de mis teoremas" - SELEC/CAT-1

**Como:** Usuario, Admin o AdminTeoremas.

**Quiero:** Poder seleccionar algunas categorias en la vista de mis teoremas.

**Para:** Mostrar los teoremas de sólo las categorías que se seleccionaron.

**Criterios de Aceptación:**

* Las categorias a seleccionar deben aparecer en un modal.
* El estado de las categorías a mostrar es persistente, por lo que cuando se vuelva a la vista de Mis Teoremas o demostración, se visualizarán solo los teoremas de las categorías seleccionadas en la última selección (tanto desde la vista de mis teoremas, como desde la vista de demostración).
* Si al cargar la vista de Demostración, el estado es que no existe ninguna categoría seleccionada, se debe desplegar el modal de selección de categorias al momento de cargar la página, con un mensaje diciendo que no tiene categorias seleccionadas, que seleccione alguna.
* Al seleccionar las categorias se debe refrescar la lista de teoremas y mostrar solamente aquellos teoremas que pertenezcan a las categorías seleccionadas.
* Los teoremas demostrados en esta vista tienen la cualidad de ser clickeables y la funcionalidad al hacer click de mostrar la lista de demostraciones en un div a parte o agrupar y desagrupar teoremas. Ocultar y mostrar categorías no debe modificar la cualidad de ser clickeable ni las funcionalidades de los clicks.
* Si se ha seleccionado un teorema para ver sus demostraciones, entonces ocultar la categoría que contiene a dicho toerema, debe ocultar también el div que contiene las demostraciones. Si las categorías que se ocultan no contienen al teorema seleccionado, entonces ocultar o mostrar no debe modificar la posición de los div donde se encuentran las demostraciones.

**ID:** "Seleccionar categorías a mostrar en la vista de demostración" - SELEC/CAT-2

**Como:** Usuario, Admin o AdminTeoremas.

**Quiero:** Poder seleccionar algunas categorias en la vista de demostración.

**Para:** Mostrar los teoremas de sólo las categorías que se seleccionaron.

**Criterios de Aceptación:**

* Las categorias a seleccionar deben aparecer en un modal.
* El estado de las categorías a mostrar es persistente, por lo que cuando se vuelva a la vista de Mis Teoremas o demostración, se visualizarán solo los teoremas de las categorías seleccionadas en la última selección (tanto desde la vista de mis teoremas, como desde la vista de demostración).
* Si al cargar la vista de Demostración, el estado es que no existe ninguna categoría seleccionada, se debe desplegar el modal de selección de categorias al momento de cargar la página, con un mensaje diciendo que no tiene categorias seleccionadas, que seleccione alguna.
* Al seleccionar las categorias se debe refrescar la lista de teoremas y mostrar solamente aquellos teoremas que pertenezcan a las categorías seleccionadas.
* Los teoremas de esta vista tienen diferente cualidades y funcionalidades al hacerles click y estas dependen de la etapa en que se encuentre el proceso de demostración y el método de demostración escogido. El estado en que se encuentre la lista de teoremas con respecto a la etapa de demostración no debe cambiar si se oculta y luego se muestra una categoría. (Para validar esto se debe listar todos los posibles estados en que puede estar la lista de teoremas, y luego ocultar y mostrar las categorias, en los diferentes escenarios en que ocurran todos los estados posibles de la lista de teoremas).
* Ocultar o mostrar categorías no debe modificar la maquetación de la página.

**ID:** "Guardar el grafo resultante después de una corrida, con todos los términos predefinidos de la versión al momento de la corrida " - VIS/REND-6

**Como:** Usuario

**Quiero:** poder guardar el grafo resultante de una corrida, con todos los términos predefinidos al momento de la corrida.

**Para:** tener la corrida y su configuración siempre disponible en cualquier momento.

**Criterios de Aceptación:**

* Efectivamente debe guardar el grafo resultante de la corrida.
* La acción debe realizarse de forma intuitiva.
* Debe mostrar un mensaje de notificación.
  + 1. Visualizar estudiantes

**ID:** "Listar materias y estudiantes" – LIST/EST-1

**Como:** Admin

**Quiero:** Poder listar las materias

**Para:** ver los estudiantes asociados a cada materia.

**Criterios de Aceptación:**

* La vista donde se pueden listar las materias, solo se puede acceder por medio de un enlace en el menú, que está visible solo si eres admin. Los usuarios que no son admin no pueden ver este enlace.
* En la lista de materias, cada materia debe ser clickeable para así desplegar una lista estudiantes inscritos en esa materia.
* Las listas de estudiantes de una materia no debe aparecer como lista de estudiantes de otra materia con distintos estudiantes.
* Cada estudiante debe ser clickeable, de modo que un click lleve a la vista de los teoremas del estudiante.
* Deben aparecer todos los teoremas del estudiante y nada más los del estudiante.

**ID:** "Listar teoremas de estudiante" – LIST/EST-2.

**Como:** Admin

**Quiero:** Poder listar los teoremas de un estudiante

**Para:** Poder ver sus demostraciones.

**Criterios de Aceptación:**

* Deben aparecer los teoremas del estudiante indicando cuales de ellos el estudiante ha demostrado.
* Haciendo click a un teorema demostrado del estudiante, se deben poder ver la lista de demostraciones del teorema de ese estudiante.
* El menú superior y footer de la vista, debe ser la del admin y no la del estudiante.
* Usar la funcionalidad de ocultar y mostrar categorias en esta vista del admin para inspeccionar al estudiante, no debe cambiar las categorias visibles de la sesión del usuario
  + 1. Regla de Leibniz Usando el Mouse

**ID:** "Subrayar subfórmula a reemplazar" – REEMP-1.

**Como:** usuario, Admin

**Quiero:** poder subrayar una subfórmula en la última línea de una demostración.

**Para:** Indicar que esa subfórmula va a ser reemplazada.

**Criterios de Aceptación:**

* La última línea de una demostración debe capturar el evento de subrayado, de forma que subrayar una subfórmula, debe disparar un trigger en el que se escribe automáticamente la expresión de regla le Leibniz en el campo correspondiente.
* Solo la última línea de una demostración captura los eventos de subrayado.
* Subrayar una subfórmula mal formada debe capturar a la subfórmula bien formada más pequeña que contenga a la fórmula mal formada.
* Cambiar la notación de los símbolos no debe afectar la funcionalidad de subrayado.
* Símbolos que no sean los preestablecidos, se deben imprimir en el campo de Leibniz, en notación funcional con las constantes representadas en el formato C#num
* Se debe probar esta funcionalidad con simbolos de diferentes aridad (1, 2 y 3), para validar que el concepto de fórmula bien formada que reconoce el algoritmo, no está fijo para aridad 1 y 2 solamente.
* Al subrayar una subfórmula que contiene una abreviación, se debe imprimir lo mismo en el campo de Leibniz, independientemente de si al momento de subrayar, se estaba mostrando la abreviación o se estaba mostrando la fórmula que se esta abreviando.
* Si existe una abreviación en la última línea de una demostración y esta queda fuera del alcance de lo subrayado, entonces se debe imprimir en el campo de Leibniz, lo fórmula con las abreviaciones y no con la fórmula que abrevia. Esto es independiente de si al momento de subrayar se estaba mostrando la abreviación o la fórmula que se estaba abreviando.
* Si existe una abreviación en la última línea de una demostración y se muestra la fórmula abreviada haciendo click sobre la abreviación, para luego subrayar un sector de esta fórmula, entonces se debe imprimir en el campo de Leibniz la fórmula correspondiente pero considerando como si la abreviación en cuestion no existiera.
  1. Introducir Simbolos

**ID:** "Introducir Símbolo predeterminado" – INTRO/SIM-1.

**Como:** AdminTeoremas

**Quiero:** Guardar simbolos predeterminados.

**Para:** Que estén disponibles para todos los usarios a la hora de construir una fórmula.

**Criterios de Aceptación:**

* El formulario debe contener los campos simbolo en latex, aridad, precedencia, notación y opcionalmente el campo infijo.
* Al seleccionar el campo infijo aparecen los campos asociación a izquierda, asociación a derecha, y desaparecen los campos de aridad y notación. (se pueden deducir de los anteriores.
* Si no es infijo, no se debe dejar el campo de aridad y precedencia vacíos.
* Si se introduce un símbolo, este símbolo está visible para cualquier usuario

**ID:** "Introducir Símbolo propio" – INTRO/SIM-2.

**Como:** usuario

**Quiero:** Guardar símbolos propios.

**Para:** Que estén disponibles, sólo dentro de la sesión del usuario, para construir una fórmula.

**Criterios de Aceptación:**

* El formulario debe contener los campos simbolo en latex, aridad, precedencia, notación y opcionalmente el campo infijo.
* Al seleccionar el campo infijo aparecen los campos asociación a izquierda, asociación a derecha, y desaparecen los campos de aridad y notación. (se pueden deducir de los anteriores.
* Si no es infijo, no se debe dejar el campo de aridad y precedencia vacíos.
* Si se introduce un símbolo, este símbolo está visible dentro de lista de símbolos del usuario.
* Simbolo que introduce un usuario no debe estar visible en la sesión de otro usuario.
  1. Modificar Símbolos
     1. **ID:** "Modificar símbolo predeterminado" – MOD/SIM-1.

1. **Como:** AdminTeoremas
   * 1. **Quiero:** Modificar simbolos predeterminados.
2. **Para:**  Modificarlos a todos los usarios a la hora de construir o visualizar una fórmula.
3. **Criterios de Aceptación:**

* El formulario debe contener los campos simbolo en latex, precedencia, notación y opcionalmente el campo infijo. Aridad no se puede modificar.
* Al seleccionar el campo infijo aparecen los campos asociación a izquierda, asociación a derecha, y desaparece el campo de notación (se pueden deducir de los anteriores).
* Si no es infijo, no se debe dejar el campo de precedencia vacío.
* Al modificar un símbolo la presentación de las fórmulas que lo contienen cambian para todos los usuarios, según lo que indica el campo de notación, simbolo en latex, etc.
  + 1. **ID:** "Modificar símbolo propio" – MOD/SIM-2.

1. **Como:** usuario
   * 1. **Quiero:** Modificar simbolos propios.
2. **Para:**  Que se modifiquen solo en mi sesión a la hora de construir o visualizar una fórmula.
3. **Criterios de Aceptación:**

* El formulario debe contener los campos simbolo en latex, precedencia, notación y opcionalmente el campo infijo. Aridad no se puede modificar.
* Al seleccionar el campo infijo aparecen los campos asociación a izquierda, asociación a derecha, y desaparece el campo de notación (se pueden deducir de los anteriores).
* Si no es infijo, no se debe dejar el campo de precedencia vacío.
* Al modificar un símbolo la presentación de las fórmulas que lo contienen cambian para solo el usuario, según lo que indica el campo de notación, simbolo en latex, etc.
  1. Eliminar Símbolos
     1. **ID:** "Eliminar símbolo predeterminado" – MOD/SIM-1.

1. **Como:** AdminTeoremas
   * 1. **Quiero:** Eliminar simbolos predeterminados.
2. **Para:**  Que no esten disponibles para ningún usuario a la hora de construir una fórmula.
3. **Criterios de Aceptación:**

* Al eliminar un símbolo ya no se encuentra disponible para ningún usuario a la hora de construir una fórmula.
* Al eliminar un símbolo que era usado en una fórmula de algún usuario, este símbolo se imprime con el formato C#id y en notación funcional, donde #id es el número del id del símbolo que se borró. El símbolo debe ser clickeable, de modo que lleve a la vista de insertar símbolo y poder recuperar los campos del símbolo con la misma aridad que tenía inicialmente.
  + 1. **ID:** "Eliminar símbolo propio" – MOD/SIM-2.

1. **Como:** usuario
   * 1. **Quiero:** Eliminar simbolos propios.
2. **Para:**  Que no esten disponibles para el usuario a la hora de construir una fórmula.
3. **Criterios de Aceptación:**

* Al eliminar un símbolo ya no se encuentra disponible para el usuario a la hora de construir una fórmula.
* Al eliminar un símbolo que era usado en una fórmula del usuario, este símbolo se imprime con el formato C#id y en notación funcional, donde #id es el número del id del símbolo que se borró. El símbolo debe ser clickeable, de modo que lleve a la vista de insertar símbolo y poder recuperar los campos del símbolo con la misma aridad que tenía inicialmente.
  1. Demostrar
     1. **ID:** "demostrar" –DEM.

1. **Como:** usuario
   * 1. **Quiero:** Seleccionar un teorema
2. **Para:**  Para realizar una demostración.
3. **Criterios de Aceptación:**

* Debe aparecer una lista con teoremas clickeables si no son axiomas. Cada axioma o teorema que ya esté demostrado por lo menos una vez, tiene un candado abierto, del resto tienen candado cerrado.
* Al seleccionar un teorema cuya demostración se había dejado a mitad, debe aparecer un mensaje que indique que va a continuar esa demostración pendiente.
* Si no hay una demostración pendiente del teorema al hacer click se debe navegar a una vista en la que se pueda seleccionar el método a usar en la demostración.
* Si el teorema seleccionado no le aplica un método de demostración, entonces la opcion de ese método debe estar inhabilitado.

* 1. Método Directo
     1. **ID:** "Método directo" –MD.

1. **Como:** usuario
   * 1. **Quiero:** Seleccionar el método directo
2. **Para:**  Para realizar una demostración.
3. **Criterios de Aceptación:**

* La opción de seleccionar el método de demostración solo aparece luego de seleccionar el teorema a demostrar.
* Luego de seleccionar el método directo deben aparecer todos los teoremas demostrados, axiomas y el teorema a demostrar listados y clickeables para poder seleccionar el enunciado de partida del método.
* Si esta es una segunda demostración del mismo teorema se deben eliminar de la lista de teoremas aquellos donde todas sus demostraciones, usen de forma directa o indirecta el enunciado a demostrar (esto se puede calcular con un algoritmo de punto fijo partiendo como conjunto inicial los axiomas, luego los teoremas que se demuestran de los axiomas directamente y así).
* Al seleccionar alguno de los teoremas o axiomas del item anterior, debe copiarse como primera línea de la demostración, el enunciado seleccionado. A partir de este momento debe estar visible el formulario para hacer inferencias y se deben listar solo los teoremas demostrados y los axiomas (el enunciado del teorema a demostrar ya no debe estar listado entre los enunciados disponibles).
  1. Inferir
     1. **ID:** "Inferir" –INFER.

1. **Como:** usuario
   * 1. **Quiero:** Hacer un paso de inferencia
2. **Para:**  Para completar una demostración.
3. **Criterios de Aceptación:**

* Hay que poner todas las combinaciones cuando dejas los campos vacios o los rellenas incorrectamente.
* Si existe una abreviación fuera del sector que se reemplaza por medio de la regla de Leibniz, en la línea siguiente luego de la inferencia se debe mantener igualmente la abreviación en el mismo sector de la fórmula.
* Luego de una inferencia todas las abreviaciones se muestran ocultando la fórmula real que abrevia, esto es independiente a si el usuario antes de la inferencia, hizo click sobre alguna abreviación mostrando la fórmula oculta.
  1. Ingreso de Formulas Asistido
     1. **ID:** "Ingreso de Formulas Asistido" –IFA.

1. **Como:** usuario
   * 1. **Quiero:** Ingresar formulas por medio de botones.
2. **Para:**  Para disminuir la posibilidad de error.
3. **Criterios de Aceptación:**

* Se debe mostrar un botón que dice “Available Symbols” tal que al hacer click sobre el se debe desplegar una lista de botones para cada símbolo disponible.
* Al hacer click en cualquiera de los botones proporcionados en “Available Symbols”, si la sección de la formula se encuentra vacía se debe insertar la operación y si dicha operación tiene argumentos (no es una constante) también deben aparecer una cantidad n de elementos de input donde n es la cantidad de argumentos que recibe el operador.
* Si se fija el cursor en alguno de los elementos de input generados anteriormente y mientras el cursor se encuentra en alguno de esos elementos se hace click sobre uno de los botones mencionados anteriormente, se reemplazará el cuadro de input sobre el cual estaba el cursor por una nueva expresión tal como se menciono en el punto anterior.
* Si se intenta presionar la tecla de delete sobre un cuadro de input vacío generador por esta funcionalidad entonces se borrara toda la expresión relacionada al operador padre de ese input y se reemplazará con un nuevo elemento de input, por ejemplo supongamos que en este momento se tiene la siguiente expresión 3 + ( \_ + ( 4 – x ) ) donde \_ representa un elemento de input vacío, si se pone el cursor sobre ese elemento y se presiona delete la expresión resultante será 3 + \_ .
* En los elementos de input generados por esta funcionalidad solo se puede escribir un carácter ya que las variables deben ser un carácter.
* Si se escribe en un elemento de input una variable y luego en alguna parte de la expresión se generan elementos nuevos de input o se borran expresiones que no eliminen el elemento de input con la variable entonces se mantiene escrita esa variable, es decir no desaparece.
* Al presionar el botón de “Clean” se debe vaciar el recuadro de formulas para que le usuario pueda volver a comenzar a escribir.