

Informe sobre integralces

fecha: 10 de julio 2015

Indice

- Estandarización
- 2332715-ICES statistics help doesn't link correctl
- 2208065-Error accessing /node/add
- 2332189-PHP Fatal error in ces_bank.logic.inc crea
- 2328287-Php notice in ces_summaryblock_form_submit
- 2314763-php notice in user/register/XXXX
- Vista para ces_message
- 2348383-Edit Issue Broken links in message list
- 2208065-Error accessing /node/add II
- Fallo en test con ces_message
- **Script documentación**
 - Documentación interna en Script documentación
- 2377529-Some translatable words
- Importación de ecoxarxe
- Add Wants X Offers without login
- Errores que se muestran al no ser usuarios integra
- Expandir tests en bank
- 2373325-Add help text to "account users" page
- **Nuevas traducciones**
 - Documentación interna en Nuevas traducciones
- Documentar transacciones externas
- 2411453-Found anonymous user as account manager
- Permisos sobre usuarios a administradores de redes
- Migración de ecoemporda
- 2421653-The format in wants.csv was changed
- createdoc idiomas automaticos
- Full project
- Añadir reply-to a mensajes
- Anular una transaccion
- Repasando código
- Nueva versión
- #2463543-Unlock users from network administrator
- Incidencias con usuarios

- URGENT: Incidències altes socis autoocupats
- Dar de baja a usuarios dados de alta por error
- Total general

Estandarización

Comprobar estandarización.

2332715-ICES statistics help doesn't link correctl

Help in admin/help/ces_statistics doesn't link correctly to the jplot library.

2208065-Error accessing /node/add

Error accessing /node/add:

Fatal error: Unsupported operand types in /var/www/ces/modules/system/system.module on line 2128 But is possible to access directly to the "Create Article" form in /node/add/article or "Create Basic page" form in /node/add/page. Something's wrong in the middle step (list of links to the diferent content types forms)

As reported here: <https://www.drupal.org/node/2267557#comment-9137293> this error is still alive, and it seems to be annoying for this user.

Is it really a drupal bug? No installation of mine fails like this... Sorry for reactivating the issue if it's a drupal bug.

2332189-PHP Fatal error in ces_bank.logic.inc crea

When a new exchange is created and the module ICES Message is NOT enabled, there's a fatal error: Fatal error: Call to undefined function ces_message_send_notify() in ices/ces_bank/ces_bank.logic.inc on line 50

2328287-Php notice in ces_summaryblock_form_submit

Bug seen in www.integralces.net: Notice: Undefined index: account en ces_summaryblock_form_submit() (línea 334 de ices/ces_summaryblock/ces_summaryblock.module).

when a logged in user access to several pages:

- /ces/bank/account/message
- /user
- /node/5

2314763-php notice in user/register/XXXX

When anonymous user access to user/register/CICE?lang=es it reports:

Notice: Undefined index: default_lang a ces_bank_form_alter() (line 1191 ices/ces_bank/ces_bank.module).

When anonymous user access to user/register/EXBA?lang=es it reports:

Notice: Undefined index: default_lang a ces_bank_form_alter() (line 1191 ices/ces_bank/ces_bank.module).

Vista para ces_message

Generar vistas con ces_message.

2348383-Edit Issue Broken links in message list

The links to single messages from the message list at ces/bank/account/message are broken. The problem is that the id in the links has a comma as thousands separator due to number formatting that should be disabled.

Example::

```
<a href="/drupaltest/en/ces/bank/account/message/16%2C515/view">[Ecoxarxa del Maresme] Your account LAIECOOP has been credited</a>
```

Tambien veo que hay mensajes que se repiten.

2208065-Error accessing /node/add II

The bug is solved, but now the navigation menu has a new item 'Add new content'. It mustn't appear.

Fallo en test con ces_message

Hola Edu!

Estava a punt de pujar un petit desenvolupament i veig que no em passen els tests del ces_message (funció testCesMessage()). M'estranya que sigui degut als canvis que he fet jo, així que volia demanar-te si a tu sí que et passen.

Merci!
Esteve

Script documentación

Creamos un script para generar la documentación con doxygen en los diferentes idiomas.

Documentación interna en Script documentación

docs/es/readme.dox:

```
@file
@brief Documentación para desarrolladores
@mainpage

# Documentación para desarrolladores

Bienvenido o bienvenida a la documentación para desarrolladores del proyecto
IntegralCES. Ésta es una documentación permanentemente en construcción que
vamos completando a medida que nos es necesario.

@section about Acerca de

Integral CES es un conjunto de módulos de Drupal para la gestión de monedas sociales
y sus comunidades.
```

ver [integralces.net](http://www.integralces.net) <<http://www.integralces.net>> para obtener más documentación y demo.integralces.net <<http://demo.integralces.net>> para un sitio de demostración.

@section alta_exchange Dar de alta nuevo exchange

Visitar el link 'New exchange'. Rellenar el formulario múltiple y enviarlo. El administrador lo activará, con lo que daremos de alta el grupo de intercambio y el usuario drupal que necesitas, con una cuenta vinculada al grupo de intercambio.

A partir de que demos de alta el exchange y tu usuario administrador, los otros usuarios de la eco-xarxa se tendrán que dar de alta usando el selector de la home (New to IntegralCES?), seleccionando el país, i pidiendo una cuenta en el exchange que acabamos de dar de alta. Esas peticiones de usuarios ya tendrían que llegar al administrador, Se activan y listo!

@section modulo_raiz Módulo raíz CES

El módulo de la raíz se denomina ICES. Este módulo por sí solo no proporciona ninguna función para el usuario final. Tiene un par de objetivos:

- Se trata de un contenedor para todos los demás módulos en una carpeta principal por lo que facilitar la instalación y actualización.
- Proporcionar algunas bibliotecas compartidas para ser utilizados por cualquiera de los submódulos.

Los archivos que pertenecen directamente al módulo de raíz son incluyen la `ces.info` obligatorio y `ces.module` y las bibliotecas compartidas. Actualmente hay una biblioteca compartida. Consiste en un capa de abstracción de base de datos ubicada en el archivo `commons.db.logic.inc`.

Base de datos

Hay una forma común para acceder a la base de datos de Drupal en todos los sub-módulos CES. Se trata de un versión aproximada del [patrón de registro activo](http://en.wikipedia.org/wiki/Active_record_pattern). El objetivo de esta capa es concentrar toda la el trabajo de base de datos en un solo lugar estandarizada. Aísla a otro código de la capa de base de datos de Drupal y aquellos que facilita la detección de errores, permite un sistema de captura centralizada y hace que la adaptación a API Drupal cambia casi trivial. Esta interfaz común se define en el archivo `commons.db.logic.inc`.

Este archivo define varias clases:

- `IcesDBObject`: La clase abstracta que cualquier objeto persistente debe extenderse. Por ejemplo, la clase `Exchange` extiende `IcesDBObject`. Para cada clase se extiende `DBObject` hay una tabla en la base de datos. Esta clase tiene una propiedad crucial llamada `table`. Una clase hija debe establecer esta variable para el nombre de tabla asociado con él. También tienen implementaciones por defecto para los dos métodos que convierten las variables de objeto interno en un array asociativo se salven y viceversa. Por objeto predeterminado propiedades públicas se convertirán en campos de la tabla.

- IcesSerializer: Esta clase tiene métodos para guardar, actualizar, cargar y eliminar cualquier objeto de tipo IcesDBObject. Es el puente real con la capa de base de datos de Drupal. Más allá de la interface CRUD básica, que ofrece algunos otros métodos de ayuda que también cuenta con una serialización especial estrategia para las jerarquías de clases que implementan el [patrón decorador](https://es.wikipedia.org/wiki/Decorator_%28patr%C3%B3n_de_dise%C3%B1o%29).
- IcesLockSerializer: Esta es una subclase de IcesSerializer para aquellas operaciones que son delicadas debido a la concurrencia. Esta clase asegura que cuando se carga un objeto, que no será cargado por cualquier otro LockSerializer hasta que lo guarde.
- DbTransaction: Un contenedor vacío para el módulo CesBank transaction.

![Base de datos](../img/database.png)

@section modulo_ces_bank Módulo CES BANK

Este es el módulo bancario principal. Define y trabaja con todos los objetos de banca: grupos de intercambio (exchnages a partir de ahora), cuentas, transacciones ... Este módulo se divide en dos partes:

- La API de banco (@subpage ces_bank.logic.inc) es el responsable de hacer toda la lógica de la banca. Se encuentra en el fichero bank.logic.inc. Cuenta con una interfaz pública para crear cuentas, hacer transacciones, etc
- El módulo de banco (@subpage ces_bank) es propiamente el módulo de Drupal con su .info y archivos module y otros archivos de páginas y formularios. Utiliza la API de banco para realizar las acciones ordenadas por la interfaz de usuario web.

La API del banco

La API del banco contiene toda la lógica de la gestión de cuentas. Se organiza en un conjunto de clases. Hay sólo una clase pública que actúa al mismo tiempo como un controlador para el conjunto y como interfaz publica. Todas las otras clases son privadas y no deben utilizarse desde fuera. Incluso los parámetros públicos de la clase Bank no están accesibles para los otros los tipos de clases. La API banco debe ser pensada como una biblioteca externa. En particular, es independiente de Drupal. Por la forma en que podría ser utilizado en integraciones dentro de otras plataformas. Utiliza las funciones de acceso a base de datos desde el módulo raíz CES y utiliza las funciones de mensajería del módulo @subpage ces_message. A continuación se comenta el director clases y sus atribuciones.

- CesBank: Interfaz pública para todas las funciones previstas para esta API. Los parámetros y el regreso Los valores de esta clase son, cuando se necesitan datos estructurados, arrays asociativos. El trabajo es diferido a los objetos particulares concretos. Una implementación típica función de esta clase es recibir y (mínimamente) analizar la entrada, cree el objeto privado apropiado, llamar a este objeto de función adecuada, (mínimamente) analizar el resultado, y devolverlo a la persona que llama.
- CesBankExchange: Representa un grupo de intercambio de comercio. Tiene un nombre, un código de 4 letras, y la cúpula opciones configurables.

- `CesBankAccount`: Es una interfaz que representa una cuenta bancaria. Actualmente sólo hay una implementación de esta interfaz. Se espera para el futuro utilizar esta interfaz para recoger cuentas a distancia.
- `CesBankLocalAccount`: Implementación de la cuenta para las cuentas almacenadas en la base de datos local. Una cuenta pertenece a un intercambio particular. Cada uno tiene un identificador único, comenzado con el código de 4 letras de su intercambio, seguido de 4 números. Especial cuentas pueden terminar con 4 letras en lugar de números. Cada `LocalAccount` es potencialmente relacionadas con uno o más usuarios, y cada usuario puede tener uno o más cuentas de este tipo. Cada cuenta tiene también un `LimitChain`, que determina la límites para esta cuenta especial de crédito y débito (y otros).
- `CesBankAccountLimit`: Límite abstracto Atómica para una cuenta. Es la piedra angular de `LimitChains`.
 - `AbsoluteDebitLimit`: Limita el débito total de una cuenta.
 - `AbsoluteCreditLimit`: Limita el crédito total de la cuenta.
- `CesBankLimitChain`: Un conjunto ordenado de `AccountLimit`, identificado con un nombre. Cada cuenta tiene un enlazar con un objeto de este tipo. Una cuenta satisface una `LimitChain` si satisface todas sus `CesBankAccountLimits`. Un intercambio puede configurar varios `CesBankLimitChains` para varios tipos de cuentas. Por ejemplo, un 'default' para los nuevos miembros, una "organización" para las cuentas compartidas que necesitan más flexibilidad y una "pública" para las cuentas públicas sin ningún tipo de límite.
- `CesBankTransactionInterface`: Interfaz para la transacción entre dos cuentas. Tiene métodos para acceso a ambos códigos de cuenta, la cantidad y la descripción o el concepto de la transacción. Existen varias implementaciones utilizando un esquema de patrón decorador.
 - `CesBankBasicTransaction`: Transacción entre dos cuentas del mismo intercambio.
 - `CesBankDecoratedTransaction`: Resumen de transacciones que delega todos los métodos para una transacción "padre" que contiene como una propiedad. Esta clase puede heredar añadir funciones a una transacción.
 - `CesBankInterExchangeTransaction`: Hijo de `DecoratedTransaction` para permitir el comercio entre cuentas de distintos intercambios.

ciclo de vida de una transacción.

Las transacciones tienen sólo dos operaciones: aplicar una nueva transacción y revocar una aplicada anteriormente. Sin embargo, una transacción pasará por varios estados entre que se crea y pasa a ser efectiva e irreversible.

Este ciclo de vida se resume en el siguiente diagrama que ahora será comentado estado por estado.

- `NEW`: Estado inicial. La transacción ha sido creada pero aún no se ha activado. Este es el estado mientras la transacción puede ser editada. A la espera de que el usuario confirme la transacción.

- TRIGGERED: La transacción esta a la espera de ser aceptada. Inmediatamente después de que se pide aceptación por parte de todas las autoridades pertinentes. Se está a la espera de su respuesta. Se pasará a ACCEPTED o REJECTED.
- ACCEPTED: La transacción ha sido aceptada por todas las autoridades. Este estado tiene una vida muy corta en las transacciones locales. El siguiente estado es ERROR si el sistema no aplica la transacción, por ejemplo, un servidor remoto no pudo responder, o COMMITED.
- COMMITTED: La transacción se aplicó con éxito. Sin embargo, puede ser revocada. Si hay no son temas para esta transacción después de un tiempo prudencial, que se archivará.
- ARCHIVED: La transacción está archivada y aplicada con éxito. Nada se puede hacer con ella. Existe sólo como datos históricos.
- REJECTED: La transacción, previamente disparada para ser aplicada, ha sido rechazada por algunos de las autoridades. Se desecha después de un tiempo prudencial. Puede pasar a ACCEPTED si algunas autoridades rectifican.
- DISCARDED: La transacción se descarta y no se ha aplicado. Nada se puede hacer con ella. Existe sólo como datos históricos.
- ERROR: Se produjo un error del sistema al aplicar o revocar la transacción. El administrador debe tener cuidado de él.

Estos fueron los distintos estados que una transacción puede pasar, A continuación los estados relacionados con la operación de revocación. El estado inicial de la operación de revocación es el de COMMITED.

- REVOKE TRIGGERED: La transacción se ha disparado paraa ser revocada. Esperando la aceptación de la operación por parte de todas las autoridades.
- REVOKE ACCEPTED: La operación de revocación ha sido aceptada por todas las autoridades. El sistema está revocando efectivamente la transacción. Este estado es muy efímero en transacciones locales. El siguiente estado será REVOKED o ERROR si el sistema no revoca la transacción.
- REVOKE REJECTED: Algunas autoridades no aceptaron la operación de revocación. La transacción seguirá en estado COMMITED y pasará a ARCHIVED después de un tiempo prudencial.
- REVOKED: La transacción ha sido revocado con éxito. Pasara a DISCARDED después de un tiempo prudencial.

El estado de la transacción se cambia el uso de la función setstate() de la interfaz de transacción. Este función también desencadena las acciones apropiadas. Por ejemplo, cuando el estado se cambia de TRIGGERED a ACCEPTED, que inicia la aplicación efectiva de la transacción.

![Estados de las transacciones](../img/states_transaction.png)

CES Banco diagrama de clases del módulo

Diagrama de clases para el módulo bancario. Es una aproximación a la implementación real: hay muchos más propiedades y métodos, y algunos de ellos pueden tener nombres distintos. Sin embargo, la esencia arquitectónica es correcta.

```
![Diagrama de clases](../img/class_diagram.png)
```

```
### Algunos componentes de IntegralCES
```

- @subpage install
- @subpage flujo_trabajo
- @subpage ces_permissions_doc
- @subpage external_transactions_doc

2377529-Some translatable words

Some translatable words in source messages code are not in english

Posted by esteve.badia on November 18, 2014 at 5:55pm

Some words in the source files in the message module are not in English. See the words "següent" and "últim" in the attached screenshot.

Importación de ecoxarxe

Ja tinc les dades de la xarxaECO i estic fent una prova d'importacio en local. Els passos 1 i 2 cap problema. En el pas 3, acaba amb aquest error: La fila 218 no s'ha pogut parsejar.

Si li dic "Salta fila", fa altre cop el procés 3 però acaba donant errors en les següents files: L'usuari no s'ha trobat a la fila per importar 219. Serà un usuari d'una altra xarxa d'intercanvi encara no importada. >> salta fila L'usuari no s'ha trobat a la fila per importar 220. Serà un usuari d'una altra xarxa d'intercanvi encara no importada. >> salta fila L'usuari no s'ha trobat a la fila per importar 221. Serà un usuari d'una altra xarxa d'intercanvi encara no importada. etc ...

Miro aquestes files als csv i no veig cap cosa estranya. El cas és que no deixa passar del pas 3 Als botons encara surt lo de @tipus d'importació, o sigui que no veig ben bé què està important (offers, wants, etc...)

Teniu idea de quin pot ser l'error? Voleu que us passi les dades de la xarxaECO per poder mirar-ho?

Add Wants X Offers without login

When not logged in, an anonymous user shouldn't be able to access the forms to add wants or offers. However in the default installation these pages are presented and accessible, in concrete ces/bank/account/myoffers and ces/bank/account/mywants.

Errores que se muestran al no ser usuarios integra

```
Notice: Undefined index: account a ces_summaryblock_form_submit() (línia 334 de /home/pr
PDOException: SQLSTATE[23000]: Integrity constraint violation: 1062 Duplicate entry '' f
```

Expandir tests en bank

Acabar de generar los test para ces_bank.

2373325-Add help text to "account users" page

Añadir el texto de ayuda a la página de "usuarios de la cuenta". De hecho, no está claro lo que significa la casilla administrador.

Nuevas traducciones

Ok, merci.

Por si te sirve, te envío lo que podría ser la cabecera en inglés, que corresponde con la ayuda del módulo.

Salut!

About

Integral CES is a suite of Drupal modules featuring social currencies management for communities. See integralces.net <<http://www.integralces.net>> for more documentation and demo.integralces.net <<http://demo.integralces.net>> for a demonstration site.

Initializing an exchange group

Enable the sub-modules you need

/CES Bank/ is needed to run Integral CES, is about management of exchange groups, users accounts, transactions between accounts, etc... */CES Notifications/* is the messaging system of Integral CES. */CES Offers and Wants/* allows users to publish their offers and wants in the exchange group. */CES user profile/* extends Drupal user profile with fields about Integral CES. */CES Summary Block/* enables a block with information about the user's account, and a switcher between accounts, if the user owns several. */CES exchange blog/* enables a blog for every exchange group and the exchange administrator can publish entries. */CES Statistics/* produces some statistics of the exchange groups.

Create new exchange group

Visit new exchange form
<<https://www.integralces.net/ces/bank/exchange/new>> and initialize

the first exchange group. You must fill in the Administrator section with the user that will be the exchange administrator (It cannot be the current drupal admin).

Visit admin exchanges page

<<https://www.integralces.net/ces/admin/ces>>, click on the new exchange and activate it.

Login as exchange administrator and write some welcome blog posts.

Invite the users to create an account

Users can sing up and create a new account visiting

/user/register/exchange_name/, where exchange_name is the short name of the new exchange group (Four numbers or uppercase letters)

Publish some offers and wants, blog entries... And have fun!

Documentación interna en Nuevas traducciones

docs/en/readme.dox:

```
@file
@brief Developer documentation
@mainpage

Welcome to the developer documentation for the IntegralCES project.
This documentation is permanently under construction and we will
complete as is necessary to us.

@section about About

Integral CES is a suite of Drupal modules featuring social currencies
management for communities. See integralces.net
<http://www.integralces.net> for more documentation and
demo.integralces.net <http://demo.integralces.net> for a demonstration site.

@section alta_exchange Initializing an exchange group

Visit new exchange form <https://www.integralces.net/ces/bank/exchange/new>
and initialize the first exchange group. You must fill in the Administrator
section with the user that will be the exchange administrator (It cannot be
the current Drupal admin).

Visit admin exchanges page /ces/admin/ces , click
on the new exchange and activate it. Login as exchange administrator and
write some welcoming blog posts.

#### Invite the users to create an account

Users can sign up and create a new account visiting
/user/register/exchange_name/ , where exchange_name is the short name of the
new exchange group (Four numbers or uppercase letters)

Publish some offers and wants, blog entries... And have fun!
```

@section modulo_raiz Base module CES

The root module is called ices. This module alone does not provide any feature to the final user. It has a pair of objectives:

- It is a container for all other modules in a parent folder so ease the installation and upgrade.
- Provide some shared libraries to be used by any of the submodules.

The files that directly belong to the root module are include the mandatory ces.info and ces.module and the shared libraries. Currently there is one shared library. It consists on a database abstraction layer located in file commons.db.logic.inc.

Database

There is a common way to access Drupal database within all the CES sub-modules. It is an approximate version of the active record pattern. The objective of this layer is to concentrate all the database work in a single standardized place. It isolates other code from the Drupal database layer and those it eases bug detection, permits a centralized catching system and makes the adaptation to Drupal API changes almost trivial. This common interface enterely defined in the file commons.db.logic.inc. This file defines several classes:

- IcesDBObject: The abstract class that any persistent object must extend. For example, the class Exchange extends DBOBJECT. For each class extending DBOBJECT there is a table in the database. This class has a crucial property called table. A child class must set this variable to the table name associated with it. It also have default implementations for the two methods that convert the internal object variables into an associative array to be saved and vice-versa. By default object public properties will become table fields.
- IcesSerializer: This class has methods to save, update, load and delete any object of type DBOBJECT. It is the actual bridge with the Drupal database layer. Beyond the basic CRUD interface, it provides some other helper methods. It also features a special serialization strategy for hierarchies of classes implementing the decorator pattern.
- IcesLockSerializer: This is a subclass of Serializer for those operations that are delicate due to concurrency. This class assures that when you load an object, it won't be loaded by any other LockSerializer until you save it.
- IcesDBTransaction: An empty wrapper for Drupal transaction.

![Base de datos](../img/database.png)

@section modulo_ces_bank Module CES Bank

CES Bank is the main banking module. It defines and works with all banking objects: exchange groups (exchages from now), accounts, transactions...

This module is divided in two parts:

- The bank API is the responsible of doing all the banking logic. It is

located at the ces_bank.logic.inc. It has a public interface to create accounts, do transactions etc.

- The bank module is properly the Drupal module with its .info and .module files and other files for pages and forms. It uses the bank API to perform the actions ordered from the web user interface.

IntegralCES Bank API

The Bank API contains all the logic of account management. It is organized in a set of classes. There is only one public class that acts at the same time as a controller for the whole set and as the public interface. All other classes are private and should not be used from outside the file. Even the public parameters of Bank class aren't of any of the other those class types.

Bank API should be thought as an external library. In particular, it is independent of D

- CesBank: Public interface for all features provided for this API. The parameters and the return values of this class are, when structured data is needed, associative arrays. The work is deferred to the concrete private objects. A typical function implementation in this class is to receive and (minimally) parse the input, create the appropriate private object, call this object's correct function, (minimally) parse the result, and return it to the caller.
- CesBankExchange: Represents an exchange group of trade. It has a name, a 4-letter code, and some configurable options.
- CesBankAccount: It is an interface representing a banking account. Currently there is only one implementation of this interface. It is expected for the future to use this interface also for remote hosted accounts.
- CesBankLocalAccount: Implementation of Account for accounts stored in local database. An account belongs to a particular exchange. Every one has a unique identifier, started with the 4-letter code of its exchange followed by 4 numbers. Special accounts can end with 4 letters instead of numbers. Every LocalAccount is potentially related to one or more users, and every user can have one or more accounts of this type. Each account has also a LimitChain, which determines the credit and debit (and other) limits for this particular account.
- CesBankAccountLimit: Atomic abstract limit for an account. It is the building block for LimitChains.
 - AbsoluteDebitLimit: Limits the total debit of an account.
 - AbsoluteCreditLimit: Limits the total credit of an account.
- CesBankLimitChain: An ordered set of AccountLimit, identified with a name. Every account has a link to an object of this type. An account satisfies a CesBankLimitChain: if it satisfies all its AccountLimits. An exchange can configure several LimitChains for several types of accounts. For example, one 'default' for new members, one 'organization' for shared accounts that need more flexibility and one 'public' for public accounts without any limit.

- CesBankTransactionInterface: Interface for the transaction between two accounts. It has methods for accessing both account codes, the amount and the description or concept of the transaction. Several implementations exist using a decorator pattern scheme.
- CesBankBasicTransaction: Transaction between two accounts of the same exchange.
- CesBankDecoratedTransaction: Abstract transaction which delegates all methods to a 'parent' transaction that contains as a property. This class can be inherited to add functions to a transaction.
- CesBankInterexchangeTransaction: Child of DecoratedTransaction to allow trading between accounts of distinct exchanges. (Still in development)

Transaction lifecycle

Transactions have only two operations: apply a new transaction and revoke a previously applied one. However, a transaction will go through several states between the creation and the effective and irreversible application. This lifecycle is summarized in the diagram below that will now be described state by state.

- NEW: Initial state. The transaction has been created but hasn't already been triggered to apply. This is the state while the transaction can be edited. Waiting for the user to apply this transaction.
- TRIGGERED: The transaction has been triggered to be applied. Immediately after it asks acceptance from all pertinent authorities. It is waiting for their response. It will become ACCEPTED or REJECTED.
- ACCEPTED: The transaction has been accepted by all authorities. It is being effectively applied by the system. This state is very short-lived in local transactions. The following state will be COMMITTED or ERROR if system fails to apply the transaction -for instance, a remote server couldn't respond.
- COMMITTED: The transaction is successfully applied. It can however be revoked. If there aren't issues for this transaction after a prudential time, it will be archived.
- ARCHIVED: The transaction is archived and successfully applied. Nothing can be done with it. It exists only as historical data.
- REJECTED: The transaction -previously triggered to be applied- has been rejected by some of the authorities. It will be discarded after a prudential time. It can still become ACCEPTED if some authorities rectify.
- DISCARDED: The transaction is discarded and has not been applied. Nothing can be done with it. It exists only as historical data.
- ERROR: Some system error occurred while applying or revoking the transaction. The administrator should take care of it.

These were the distinct states a transaction may pass through while being

applied. The following are those states related with the revoke operation. The starting state of the revoke operation is the COMMITTED state.

- REVOKE TRIGGERED: The transaction has been triggered to be revoked. Waiting for acceptance of this operation by all authorities.
- REVOKE ACCEPTED: The revoke operation has been accepted by all authorities. The system is effectively revoking the transaction. This state is very short-lived state in local transactions. The following state will be REVOKED or ERROR if system fails to revoke the transaction.
- REVOKE REJECTED: Some authorities did not accept the revoke operation. The transaction will remain applied and it will be ARCHIVED after a prudential time.
- REVOKED: The transaction has been successfully revoked. It will become DISCARDED after a prudential time.

The transaction state is changed using the setState() function of the Transaction interface. This function also triggers the appropriate actions. For instance, when the state is changed from TRIGGERED to ACCEPTED, it starts the effective application of the transaction.

![Estados de las transacciones](../img/states_transaction.png)

CES Bank module class diagram

Class diagram for the banking module. It is an approximation of the actual implementation: there are many more properties and methods, and some of them may have distinct names. However, the architectural essence is correct.

![Diagrama de clases](../img/class_diagram.png)

Other components of IntegralCES

- @subpage install
- @subpage dev_flow
- @subpage ces_permissions_doc
- @subpage external_transactions_doc
- @subpage translations_doc

docs/es/readme.dox:

@file

@brief Documentación para desarrolladores

@mainpage

Documentación para desarrolladores

Bienvenido o bienvenida a la documentación para desarrolladores del proyecto IntegralCES. Ésta es una documentación permanentemente en construcción que vamos completando a medida que nos es necesario.

@section about Acerca de

Integral CES es un conjunto de módulos de Drupal para la gestión de monedas sociales y sus comunidades.

ver [integralces.net](http://www.integralces.net) <<http://www.integralces.net>> para obtener más documentación y demo.integralces.net <<http://demo.integralces.net>> para un sitio de demostración.

@section alta_exchange Dar de alta nuevo exchange

Visitar el link 'New exchange'. Rellenar el formulario múltiple y enviarlo. El administrador lo activará, con lo que daremos de alta el grupo de intercambio y el usuario drupal que necesitas, con una cuenta vinculada al grupo de intercambio.

A partir de que demos de alta el exchange y tu usuario administrador, los otros usuarios de la eco-xarxa se tendrán que dar de alta usando el selector de la home (New to IntegralCES?), seleccionando el país, i pidiendo una cuenta en el exchange que acabamos de dar de alta. Esas peticiones de usuarios ya tendrían que llegar al administrador, Se activan y listo!

@section modulo_raiz Módulo raíz CES

El módulo de la raíz se denomina ICES. Este módulo por sí solo no proporciona ninguna función para el usuario final. Tiene un par de objetivos:

- Se trata de un contenedor para todos los demás módulos en una carpeta principal por lo que facilitar la instalación y actualización.
- Proporcionar algunas bibliotecas compartidas para ser utilizados por cualquiera de los submódulos.

Los archivos que pertenecen directamente al módulo de raíz son incluyen la `ces.info` obligatorio y `ces.module` y las bibliotecas compartidas. Actualmente hay una biblioteca compartida. Consiste en un capa de abstracción de base de datos ubicada en el archivo `commons.db.logic.inc`.

Base de datos

Hay una forma común para acceder a la base de datos de Drupal en todos los sub-módulos CES. Se trata de un versión aproximada del [patrón de registro activo](http://en.wikipedia.org/wiki/Active_record_pattern). El objetivo de esta capa es concentrar toda la el trabajo de base de datos en un solo lugar estandarizada. Aísla a otro código de la capa de base de datos de Drupal y aquellos que facilita la detección de errores, permite un sistema de captura centralizada y hace que la adaptación a API Drupal cambia casi trivial. Esta interfaz común se define en el archivo `commons.db.logic.inc`.

Este archivo define varias clases:

- `IcesDBObject`: La clase abstracta que cualquier objeto persistente debe extenderse. Por ejemplo, la clase `Exchange` extiende `IcesDBObject`. Para cada clase se extiende `DBObject` hay una tabla en la base de datos. Esta clase tiene una propiedad crucial llamada `table`. Una clase hija debe establecer esta variable para el nombre de tabla asociado con él. También tienen implementaciones por defecto para los dos métodos que convierten las variables de objeto interno en un array asociativo se salven y viceversa. Por objeto predeterminado

propiedades públicas se convertirán en campos de la tabla.

- IcesSerializer: Esta clase tiene métodos para guardar, actualizar, cargar y eliminar cualquier objeto de tipo IcesDBObject. Es el puente real con la capa de base de datos de Drupal. Más allá de la interface CRUD básica, que ofrece algunos otros métodos de ayuda que también cuenta con una serialización especial estrategia para las jerarquías de clases que implementan el [patrón decorador](https://es.wikipedia.org/wiki/Decorator_%28patr%C3%B3n_de_dise%C3%B1o%29).
- IcesLockSerializer: Esta es una subclase de IcesSerializer para aquellas operaciones que son delicadas debido a la concurrencia. Esta clase asegura que cuando se carga un objeto, que no será cargado por cualquier otro LockSerializer hasta que lo guarde.
- DbTransaction: Un contenedor vacío para el módulo CesBank transaction.

![Base de datos](../img/database.png)

@section modulo_ces_bank Módulo CES BANK

Este es el módulo bancario principal. Define y trabaja con todos los objetos de banca: grupos de intercambio (exchnages a partir de ahora), cuentas, transacciones ... Este módulo se divide en dos partes:

- La API de banco (@subpage ces_bank.logic.inc) es el responsable de hacer toda la lógica de la banca. Se encuentra en el fichero bank.logic.inc. Cuenta con una interfaz pública para crear cuentas, hacer transacciones, etc
- El módulo de banco (@subpage ces_bank) es propiamente el módulo de Drupal con su .info y archivos module y otros archivos de páginas y formularios. Utiliza la API de banco para realizar las acciones ordenadas por la interfaz de usuario web.

La API del banco

La API del banco contiene toda la lógica de la gestión de cuentas. Se organiza en un conjunto de clases. Hay sólo una clase pública que actúa al mismo tiempo como un controlador para el conjunto y como interfaz publica. Todas las otras clases son privadas y no deben utilizarse desde fuera. Incluso los parámetros públicos de la clase Bank no están accesibles para los otros los tipos de clases. La API banco debe ser pensada como una biblioteca externa. En particular, es independiente de Drupal. Por la forma en que podría ser utilizado en integraciones dentro de otras plataformas. Utiliza las funciones de acceso a base de datos desde el módulo raíz CES y utiliza las funciones de mensajería del módulo @subpage ces_message. A continuación se comenta el director clases y sus atribuciones.

- CesBank: Interfaz pública para todas las funciones previstas para esta API. Los parámetros y el regreso Los valores de esta clase son, cuando se necesitan datos estructurados, arrays asociativos. El trabajo es diferido a los objetos particulares concretos. Una implementación típica función de esta clase es recibir y (mínimamente) analizar la entrada, cree el objeto privado apropiado, llamar a este objeto de función adecuada, (mínimamente) analizar el resultado, y devolverlo a la persona que llama.
- CesBankExchange: Representa un grupo de intercambio de comercio. Tiene un

nombre, un código de 4 letras, y la cúpula opciones configurables.

- **CesBankAccount**: Es una interfaz que representa una cuenta bancaria. Actualmente sólo hay una implementación de esta interfaz. Se espera para el futuro utilizar esta interfaz para recoger cuentas a distancia.
- **CesBankLocalAccount**: Implementación de la cuenta para las cuentas almacenadas en la base de datos local. Una cuenta pertenece a un intercambio particular. Cada uno tiene un identificador único, comenzado con el código de 4 letras de su intercambio, seguido de 4 números. Especial cuentas pueden terminar con 4 letras en lugar de números. Cada LocalAccount es potencialmente relacionadas con uno o más usuarios, y cada usuario puede tener uno o más cuentas de este tipo. Cada cuenta tiene también un LimitChain, que determina la límites para esta cuenta especial de crédito y débito (y otros).
- **CesBankAccountLimit**: Límite abstracto Atómica para una cuenta. Es la piedra angular de LimitChains.
 - **AbsoluteDebitLimit**: Limita el débito total de una cuenta.
 - **AbsoluteCreditLimit**: Limita el crédito total de la cuenta.
- **CesBankLimitChain**: Un conjunto ordenado de AccountLimit, identificado con un nombre. Cada cuenta tiene un enlazar con un objeto de este tipo. Una cuenta satisface una LimitChain si satisface todas sus CesBankAccountLimits. Un intercambio puede configurar varios CesBankLimitChains para varios tipos de cuentas. Por ejemplo, un 'default' para los nuevos miembros, una "organización" para las cuentas compartidas que necesitan más flexibilidad y una "pública" para las cuentas públicas sin ningún tipo de límite.
- **CesBankTransactionInterface**: Interfaz para la transacción entre dos cuentas. Tiene métodos para acceso a ambos códigos de cuenta, la cantidad y la descripción o el concepto de la transacción. Existen varias implementaciones utilizando un esquema de patrón decorador.
 - **CesBankBasicTransaction**: Transacción entre dos cuentas del mismo intercambio.
 - **CesBankDecoratedTransaction**: Resumen de transacciones que delega todos los métodos para una transacción "padre" que contiene como una propiedad. Esta clase puede heredar añadir funciones a una transacción.
 - **CesBankInterExchangeTransaction**: Hijo de DecoratedTransaction para permitir el comercio entre cuentas de distintos intercambios.

ciclo de vida de una transacción.

Las transacciones tienen sólo dos operaciones: aplicar una nueva transacción y revocar una aplicada anteriormente. Sin embargo, una transacción pasará por varios estados entre que se crea y pasa a ser efectiva e irreversible.

Este ciclo de vida se resume en el siguiente diagrama que ahora será comentado estado por estado.

- **NEW**: Estado inicial. La transacción ha sido creada pero aún no se ha activado.

Este es el estado mientras la transacción puede ser editada. A la espera de que el usuario confirme la transacción.

- TRIGGERED: La transacción esta a la espera de ser aceptada. Inmediatamente después de que se pide aceptación por parte de todas las autoridades pertinentes. Se está a la espera de su respuesta. Se pasará a ACCEPTED o REJECTED.
- ACCEPTED: La transacción ha sido aceptada por todas las autoridades. Este estado tiene una vida muy corta en las transacciones locales. El siguiente estado es ERROR si el sistema no aplica la transacción, por ejemplo, un servidor remoto no pudo responder, o COMMITED.
- COMMITED: La transacción se aplicó con éxito. Sin embargo, puede ser revocada. Si hay no son temas para esta transacción después de un tiempo prudencial, que se archivará.
- ARCHIVED: La transacción está archivada y aplicada con éxito. Nada se puede hacer con ella. Existe sólo como datos históricos.
- REJECTED: La transacción, previamente disparada para ser aplicada, ha sido rechazada por algunos de las autoridades. Se desecha después de un tiempo prudencial. Puede pasar a ACCEPTED si algunas autoridades rectifican.
- DISCARDED: La transacción se descarta y no se ha aplicado. Nada se puede hacer con ella. Existe sólo como datos históricos.
- ERROR: Se produjo un error del sistema al aplicar o revocar la transacción. El administrador debe tener cuidado de él.

Estos fueron los distintos estados que una transacción puede pasar, A continuación los estados relacionados con la operación de revocación. El estado inicial de la operación de revocación es el de COMMITED.

- REVOKE TRIGGERED: La transacción se ha disparado paraa ser revocada. Esperando la aceptación de la operación por parte de todas las autoridades.
- REVOKE ACCEPTED: La operación de revocación ha sido aceptada por todas las autoridades. El sistema está revocando efectivamente la transacción. Este estado es muy efímero en transacciones locales. El siguiente estado será REVOKED o ERROR si el sistema no revoca la transacción.
- REVOKE REJECTED: Algunas autoridades no aceptaron la operación de revocación. La transacción seguirá en estado COMMITED y pasará a ARCHIVED después de un tiempo prudencial.
- REVOKED: La transacción ha sido revocado con éxito. Pasara a DISCARDED después de un tiempo prudencial.

El estado de la transacción se cambia el uso de la función setstate() de la interfaz de transacción. Este función también desencadena las acciones apropiadas. Por ejemplo, cuando el estado se cambia de TRIGGERED a ACCEPTED, que inicia la aplicación efectiva de la transacción.

![Estados de las transacciones](../img/states_transaction.png)

```
#### CES Banco diagrama de clases del módulo
```

Diagrama de clases para el módulo bancario. Es una aproximación a la implementación real: hay muchos más propiedades y métodos, y algunos de ellos pueden tener nombres distintos. Sin embargo, la esencia arquitectónica es correcta.

```
![Diagrama de clases](../../img/class_diagram.png)
```

```
### Algunos componentes de IntegralCES
```

- @subpage install
- @subpage flujo_trabajo
- @subpage ces_permissions_doc
- @subpage external_transactions_doc

scripts/doxygen/readme.dox:

```
@defgroup createdoc Automate doxygen
```

Script to automate the generation of documentation with doxygen.

This generates source code documentation and add the pages of the docs folder by language specified.

The "bash script", copy to a temporary folder the files of folder is that it is the default.

Replaces existing files in the folder specified language and generates documentation.

The missing documents in a language is generated in the default language.

Script para automatizar la generación de documentación con doxygen.

```
@section use Use
```

```
./scripts/doxygen/createdoc.sh debug es '1.0'
```

- Debug: Enable debug.
- es: Language.
- 1.0 version.

```
@section configuration_createdoc Settings
```

In the header of the script you can set:

- possible_languages: Active Languages. (es en ca)
- dir_ices: Directory of source code. "\$ (dirname" \$ 0 ") /../../"
- doxygen_conf: Configuration file. "scripts/doxygen/doxygen.conf"
- extra_css: Leaf css with extra values. "scripts/doxygen/extra.css"
- doxygen_header: Header page. "scripts/doxygen/header.html" is not presented correctly,
- doxygen_footer: Matching Any Word foot. "scripts/doxygen/footer.html"
- doxygen_logo: Logo. "scripts/doxygen/logo.png" replaces the project name for the logo.

If can not find the specified files not used.

@}

Documentar transacciones externas

Les transaccions ICES <=> CES funcionen a través d'un servidor intermig, la Clearing Central (CC). Així doncs és realment ICES <=> CC <=> CES. Només poden fer aquest tipus de transaccions els comptes de xarxes de l'integralces que estiguin donades d'alta a la CC i correctament configurades. Això hauria de ser la majoria: jo he comprovat HORA i COOP. Això ho ha de configurar l'administrador de la xarxa.

Una vegada està la xarxa configurada, un usuari ha d'anar al formulari habitual d'introduir transacció. Allà, al desplegable "Xarxa d'intercanvi del comprador" hi ha d'aparèixer l'opció "Xarxa externa...". A l'escollir aquesta opció, l'interfície canvia (a través d'una crida AJAX) i el que abans era un desplegable per escollir el compte a cobrir, ara són dos camps de text. Un per posar el "ID de la xarxa" o NID (cenXXXX) (amb un link al costat per a buscar aquest NID que ens interessa), i l'altre per a posar el número de compte del comprador (p.ex. BCNAXXXX).

Els altres camps són com és habitual: descripció i quantitat.

Al crear i confirmar la transacció, es fa una crida a la CC que al seu torn informa el sistema CES. Des del punt de vista de la xarxa, aquesta transacció és com la resta de transaccions interxarxes: queda anotada una transacció entre l'usuari real de l'ICES (p.ex COOP1234) i l'usuari virtual que representa la xarxa externa (p.ex. COOPcen0123).

També com a les transaccions interxarxes, aquest usuari virtual COOPcen0123 es crearà al primer intent de transacció entre aquestes dues xarxes (que fallarà) i una vegada activat, ja funcionaran les transaccions amb aquesta xarxa. Això serveix perquè l'administrador doni el vist i plau manual d'un compte de la xarxa a operar amb una altra xarxa.

2411453-Found anonymous user as account manager

Trobat usuari anònim com a gerent de compte en un compte a integralces.net No sé com va ser assignat a aquest compte, però potser hem de revisar les proteccions en el formulari.

XaviP comentar fa aproximadament 17 hores

Sembla que no és prou validat a `ces_bank_account_user_form_validate ()` a `ces_bank / ces_bank.forms.account_users.inc`

Només ha protegit contra el nom d'usuari buit, però un administrador intercanvi informa que escrivint un nom no vàlid, l'usuari es guarda amb `uid = 0`.

Permisos sobre usuarios a administradores de redes

Permitir a los administradores de una red editar los usuarios.

Migración de ecoemporda

Migramos ecoxarxe emporda

2421653-The format in wants.csv was changed

Adaptamos código.

createdoc idiomas automaticos

No hace falta preguntar idioma, miramos los que tenemos y generamos la documentación.

Full project

El nostre estimat projecte ja té la categoria que es mereix a drupal.org.

1) La nova url és:

<https://www.drupal.org/project/ices>

Ja podem canviar-la de totes les documentacions i enllaços varis!!!

2) També ha canviat l'adreça del git. Cal executar la següent comanda al directori modules/ices:

```
git remote set-url origin [username]@git.drupal.org:project/ices.git
```

Canviant [username] pel vostre usuari. Fem-ho abans de fer nous commits!

Abraçada,
Esteve

Añadir reply-to a mensajes

Hola Tot@s!

Cuando se recibe un mensaje desde un comprador interesado en un producto, la respuesta por defecto es a "webmaster@integralces.net" y probable que les lleguen muchos pedidos ahí!

Entonces he averiguado un poco, y en /ices/ces_message/ces_message.module veo que falta un hook:

```
function hook_mail($key, &$message, $params){  
  $message['headers']['Reply-To'] = <your value>;  
}
```

Como se explica aquí:

<http://www.signalfire.co.uk/blog/reply-header-drupalmail#.VO20fbVXZxA>

Una vez que haya esto, se podría poner por ejemplo que en send_offerwant_notify de las diferentes idiomas salga:

```
$reply-to = [notify:user:mail];
```

Y así seguirán viniendo de webmaster en el From: pero al responder el

mensaje iría al Reply-to: del ofertante.

Espero que sea util.. Si quieren me lo bajo y pruebo y monto sistema local y todo pero será para un finde por ahí

Anular una transaccion

alo!

Tengo activado el pago manual.
Para anular una transaccion, me da siempre error.

Gracias, elle

Repasando código

Hacemos un repaso antes de la nueva release.

Nueva versión

El 23/03/15 a las 20:48, Esteve Badia escribió:

Hola Edu,

Que bién este análisis!!!

Si te parece me encargo de lo que tiene que ver con los errores en transacciones y tu los demás errores de mensajes y demás o alguna pequeña mejora.

Con las mejoras yo no voy a ponerme, pero si alguna te apetece adelante, aunque no tocaría nada que nos pueda retrasar mucho o introducir nuevos bugs. También, antes de hacer algo que cambie mucho la experiencia de usuario (como el formulario transacción) o la estructura del programa (como lo de los permisos) me gustaría comentarlo un poco.

Así pues yo me encargo de los bugs:

El estado de las transacciones entre redes con autorización manual puede ser incoherente al ser vista desde la red emisora y receptora.

En el selector de cuenta al crear una transacción se muestran las cuentas virtuales, se debería filtrar. Cuando estamos editando una transacción no modifica dinámicamente los usuarios del selector al cambiar de red.

Mirar porque no se envía la notificación de "Transacción rechazada" veo que la plantilla si que está. (Account debit rejected).

Espero tenerlo esta semana, digamos que el jueves por la noche. Luego podríamos hacer la versión. Te parece?

Abrazo,
Esteve

Edu:

Me quedo con estas:

- Crear una vista para las transacciones en ces/bank/account/transaction. (**Para la siguiente versión**)

- En el menú Administración la sección de editar plantillas de mensaje tiene el nombre de "lista" lo cual no es muy concreto. (**Corregido**)
- Al administrador de una red se le muestra "List all" en la sección de mensajes pero al entrar no muestra nada. (**Corregido**)
- En el título del mensaje "Autorizar transacción" estaría bien añadir el nombre del emisor. (**Hecho**)
- Borrado de mensajes, hacer que el cron borre los mensajes que sean de un año atrás. (**Implementado**)
- Buscar ofertas en todas las redes y añadir información de la localización y red de las ofertas. (**Implementado**)

edu

#2463543-Unlock users from network administrator

Es necesario repasar todo el sistema de editor de usuarios.

- Cuando estas editando un usuario se presentan las opciones de las cuentas del usuario actual, no las del usuario que estas editando.
- Desbloquear un usuario no se le permite al administrador de red, hay que mirar de que lo pueda hacer.

Incidencias con usuarios

Muchos usuarios están teniendo problemas al confundir el nombre de la cuenta con el nombre de usuario.

Creamos un script que nos permita de forma rápida ver los usuarios de una cuenta para poder pasar la información.

```
#!/bin/bash

SQL="
SELECT u.uid, u.name AS usuari, u.mail, ca.name AS compte, IF (u.status=1, 'activo', 'bl
FROM users u
LEFT JOIN ces_accountuser cau ON cau.user = u.uid
LEFT JOIN ces_account ca ON ca.id = cau.account
WHERE ca.name = "

BUSCAR=${1:-BUSCAR}
DATABASE=integralcesservidor
MYSQL_USER=root

SQL="
SELECT u.uid, u.name AS usuari, u.mail, fdcf.ces_firstname_value AS firstname, fdcs.ces_
FROM users u
LEFT JOIN ces_accountuser cau ON cau.user = u.uid
LEFT JOIN ces_account ca ON ca.id = cau.account
LEFT JOIN field_data_ces_surname fdcs ON fdcs.entity_id = u.uid AND fdcs.bundle='user'
LEFT JOIN field_data_ces_firstname fdcf ON fdcf.entity_id = u.uid

WHERE fdcs.entity_id = u.uid AND
(
fdcf.ces_firstname_value like '%${BUSCAR}%'
OR fdcs.ces_surname_value like '%${BUSCAR}%'
OR ca.name like '%${BUSCAR}%'

```

```
OR u.name like '%${BUSCAR}%'
OR u.mail like '%${BUSCAR}%'
)
"
if [ "$1" == "" ] ; then
    echo -e "$SQL"
    exit
fi

mysql -u$MYSQL_USER -p`contrasenya mysql` -e "$SQL $2" $DATABASE
```

URGENT: Incidències altes socis autoocupats

Hola Edu

mira en hacer el alta las personas de GE van introduciendo los datos como solicitud, pero pasa una cosa que no sabemos y es que no quedan pendiente de activar, pero cuando van a introducir de nuevo los datos, el sistema dice que ya están, pero no lo encontramos...que puede ser?

gracias y saludos
marta

Dar de baja a usuarios dados de alta por error

Dar de baja a usuarios dados de alta por error en CICE i error usuario asignado

Hola Edu

como puedo resolver estas cosas?

gracias
marta

Hola Marta te envío la información sobre los usuarios que me pides, para que el administrador pueda hacer las acciones pertinentes.

Con esta url el administrador del integralces puede editar el usuario que quiera.

www.integralces.net/user/IDENTIFICADOR-USUARIO/edit

Cambiando IDENTIFICADOR-USURIO por el que se quiera editar, por ejemplo:
www.integralces.net/user/3870/edit

Las cuentas ices solo las puede cambiar el administrador de cada red.

<http://www.integralces.net/ces/admin/account/NUMERO-DE-CUENTA/edit>

Cambiando NUMERO-DE-CUENTA por su correspondiente, pero debes ser el admin de su red.

IMPORTANTE

Antes de eliminar un usuario sería interesante que no tuviera una cuenta asignada con movimientos, sino la cuenta quedara colgada sin usuarios.

Se puede aprovechar un usuario e incluirlo en una cuenta y borrarlo de otra.

Espero haberte ayudado.

Nos vemos.

Edu.

Total general

Totales	
Tiempo total (H:M)	124:50
Euros / hora	6.60€
Total global	841.41€