

Documentación del archivo [cron-profitability.php](#)

Autores: Ramón y Carlos

Ubicación del archivo: [cron-profitability.php](#)

## Descripción General

Este archivo se encarga de gestionar la rentabilidad y concesión de las tareas programadas. Procesa los cálculos de rentabilidad y concesión para un proyecto e inserta los resultados en la base de datos.

## Objetivo del Archivo

El objetivo principal de este archivo es calcular la rentabilidad y concesión de los proyectos y distribuir las ganancias entre los usuarios afiliados. Los resultados se almacenan en la tabla s2b\_profit\_history de la base de datos y se muestran en pantalla.

## Funciones Utilizadas

obtener\_proyectos(\$projectID = null)

Descripción: Obtiene uno o todos los proyectos de la base de datos.

Ubicación: [functions.php](#)

obtener\_usuarios\_wid()

Descripción: Obtiene todos los usuarios afiliados ordenados por ID de manera ascendente.

Ubicación: [functions.php](#)

obtener\_cantidad\_bees(\$id, \$beeType, \$projectID, \$monthPay)

Descripción: Obtiene la suma de las cantidades de bees para un proyecto específico y un tipo de abeja específico hasta una fecha determinada.

Ubicación: [functions.php](#)

## Procesos que Engloba

### 1. Inicialización de Variables

```
global $wpdb;
```

```
$pID = intval($_GET['crons-profitability']); // obtenemos el
id del proyecto
$monthPay = date('Y-m-d H:i:s'); // Establece la fecha y hora
actual en el formato 'Y-m-d H:i:s' y la asigna a la variable
$monthPay
```

## 2. Obtención de Datos

Se obtienen los datos del proyecto y los usuarios afiliados:

```
$project = obtener_proyectos($pID);
$users = obtener_usuarios_wid();
```

## 3. Cálculo de Rentabilidad para Bees de Propiedad

Se verifica si el proyecto tiene propertyBees mayor a 0 y si el estado del proyecto es 1. Luego, se calcula la rentabilidad para cada usuario y se inserta en la base de datos:

```
if ($project->propertyBees > 0 && $project->state == 1) {
    $project->propertyYield = floatval($project->
propertyYield);
    if (!empty($users) && $project->propertyYield > 0) {
        foreach ($users as $user) {
            $getBees = obtener_cantidad_bees($user->id, 1,
$project->ID, $monthPay);
            if (!empty($getBees)) {
                $calculatedYield =
floatval(bcmul($project->propertyYield, $getBees, 2));
                $calculatedYieldRefers =
floatval(bcmul($project->propertyYieldBroker, $getBees, 2));
                $wpdb->insert($wpdb->
prefix.'s2b_profit_history', [
                    'ownerID' => $user->id,
                    'projectID' => $project->ID,
                    'beeType' => 1,
                    'beeAmount' => $getBees,
                    'beeValue' => $project->propertyYield,
                    'amount' => $calculatedYield,
                    'amountRefers' => $calculatedYieldRefers,
                    'created' => $monthPay
                ]);
            }
        }
    }
}
```

```

        echo "Usuario a pagar bee propiedad: {$user->id} <br>- Cantidad Bees: {$getBees} <br>- Cantidad a Pagar: {$calculatedYield} € <br>- Cantidad a Repartir Brokers: {$calculatedYieldRefers} <hr>";
    }
}
}
}
}

```

#### 4. Cálculo de Rentabilidad para Bees de Concesión

Se verifica si el proyecto tiene concessionBees mayor a 0 y si el estado del proyecto es 1. Luego, se calcula la rentabilidad para cada usuario y se inserta en la base de datos:

```

if ($project->concessionBees > 0 && $project->state == 1) {
    $project->concessionYield = floatval($project->concessionYield);
    if (!empty($users) && $project->concessionYield > 0) {
        foreach ($users as $user) {
            $getBees = obtener_cantidad_bees($user->id, 2, $project->ID, $monthPay);
            if (!empty($getBees)) {
                $calculatedYield = floatval(bcmul($project->concessionYield, $getBees, 2));
                $calculatedYieldRefers = floatval(bcmul($project->concessionYieldBroker, $getBees, 2));

                $wpdb->insert($wpdb->prefix.'s2b_profit_history',[
                    'ownerID' => $user->id,
                    'projectID' => $project->ID,
                    'beeType' => 2,
                    'beeAmount' => $getBees,
                    'beeValue' => $project->propertyYield,
                    'amount' => $calculatedYield,
                    'amountRefers' => $calculatedYieldRefers,
                    'created' => $monthPay
                ]);
                echo "Usuario a pagar bee concesion: {$user->id} <br>- Cantidad Bees: {$getBees} <br>- Cantidad a Pagar: {$calculatedYield} € <br>- Cantidad a Repartir Brokers: {$calculatedYieldRefers} <hr>";
            }
        }
    }
}

```

```
}  
    }  
    }  
}
```