



PROYECTO BIOMÁISCO2

Memoria Libre

CALCULO DEL SECUESTRO DE CARBONO PROYECTOS DE ABSORCION

1. Objeto

El presente documento tiene por objeto describir el contenido de los scripts (en R y Python) para el cálculo del carbono almacenado en el suelo, los productos, la bioenergía y la disminución del riesgo de incendios.

Sobre estos scripts se tendrán que hacer las modificaciones oportunas para, por ejemplo, tomar los datos de inicio del formulario completado por los usuarios, leer de geoserver valores de ciertas variables en base a las coordenadas X-Y...etc.

2. Descripción de los módulos

2.1 Lectura de la API

Este primer apartado no trata en sí mismo de explicar el funcionamiento de ningún módulo elaborado por Madera+ pero se considera necesario para definir cómo será necesario pedir los datos, a partir de los cuales se harán los posteriores cálculos, a la API de la universidad.

Así pues, será preciso marcar un marco temporal (definido por el usuario) el cual nos dirá cuántos años se quiere simular en total. Se definirá el año de inicio del proyecto para poder generar una tabla con tantas filas como años tenga el marco temporal más una (el año de inicio sería año 0 y fila 1 -aunque si es en Python se genera una columna "index" que empieza en 0-, por lo que para un marco temporal de 60 años harían falta 61 filas para llegar al año 60 desde el inicio del proyecto) y en la que a cada fila se le asigne un año determinado en una columna "Año".

Una vez establecido el marco temporal y el año de inicio, en base a los datos de la silvicultura programada (ya sea indicada por el usuario o leída de las tablas de los modelos silvícolas cuando sea el caso en la simulación de la línea base) se podrá saber cuántas veces se pide a la API que realice los cálculos.

Los datos necesarios para el funcionamiento de la API son:

- stand_json: datos de partida del rodal con:
 - id_rodal: identificador del rodal
 - Especie (sp)
 - Edad de la masa inicial (t),
 - Índice de sitio (S),
 - Altura dominante (H)
 - Número de pies por ha (N)
 - Área Basimétrica (G)
- mng_presc_json: silvicultura (claras o cortas intermedias "CI" y cortas finales ("CF")), con la siguiente información:
 - id_prog_selv: identificador del programa silvícola
 - t: edad del rodal a la que realizar la intervención (años)

- R: relación de extracción de la clara (valores entre 0.35-1) segundo en la columna “Tipo” del formulario se haya indicado lo siguiente:
 - Sistemática: R=1 (las cortas finales son por defecto siempre “Sistemáticas”).
 - Mixta: R=0.75
 - Baja: R=0.45
 - I: proporción de árboles a extraer (valores entre 0.0-1.0)
 - filas_clara_sistematica: cada cuántas filas extraer una fila en clara sistemática. Tomará siempre valor 0 dado que este aspecto ya se considera intrínsecamente al seleccionar un “Tipo” de actuación con el que obtener el valor de R.
 - Ne: número de árboles por hectárea a extraer. Dado que se establece previamente un valor de I (proporción de árboles a extraer) no es necesario y se autocompleta con el valor igual a 0.
 - Np: número de árboles por hectárea después de clara. Se autocompleta siempre con valor igual a 0 por el mismo motivo que en “Ne”.
- assort_def_json: definición de destinos en punta delgada en base al diámetro en punta delgada, con la siguiente información:
- id_conf_dest: identificador de la configuración del destino comercial de la madera
 - destino: descripción del destino comercial de la madera. Habrá siempre tres destinos, que se autocompletarán en base al “Objetivo” marcado por el usuario en el formulario de modo que, en base a tablas internas por especie y objetivo, se completarán los diámetros mínimos (dmin) y los precios (precio) para enviar a la API.
 - dmin: diámetro mínimo en punta delgada (cm)
 - precio: precio esperado de venta de la madera (€/m³). Será necesario completarlo porque si no la API da error, aunque después en los cálculos de CO₂ no se considere el balance económico.

Así, de este modo, en base a la edad “t” de inicio (stand_json) y la edad “t” a la que el programa silvícola tenga una corta final (mng_presc_json), esto es, la edad a la cual el valor de “I” sea igual a 1, se podrá saber el número de filas que devolverá la API puesto que después de dicho valor ya no quedará masa forestal. Conociendo el marco temporal, la edad de inicio y la edad de la corta final (I=1) se podrá saber cuántas veces se le pide a la API que devuelva una simulación y poder autocompletar la tabla generada internamente con tantas filas como años tenga el marco temporal más uno de la que se habló previamente.

Aún con todo esto, ha de tenerse en cuenta que, en la segunda y sucesivas peticiones, el valor t de “stand_json” ha de ser siempre igual a 1, dado que, si bien en la primera petición el usuario marca la edad de la masa, en la segunda petición la masa se renueva, considerándose que se planta en el año de la corta final y, por tanto, al año siguiente tendrá edad “t” igual a 1. Además, se leerá en el caso de la línea base el valor de “N” establecido en el modelo silvícola de que se trate. En caso de proyecto de absorción, el usuario marcará la densidad de plantación de la nueva masa tras la corta final y se usará dicho dato para enviar la solicitud a la API.

El valor de índice de sitio “S” de stand_json se tomará del valor indicado por el usuario para la primera simulación, de modo que se considera que el índice de sitio es propio del lugar. El cálculo del índice de sitio según la especie se incluye en el script “Solicitudes.py”.

De este modo se expone un ejemplo de envío de datos a la API en la primera y en las sucesivas simulaciones:

```
1) stand_json <- '{"id_rodal":"rodal_1","sp":28,"t":5,"S":20,"H":3.7149,"N":1200,"G":40}'
2) stand_json <- '{"id_rodal":"rodal_1","sp":28,"t":1,"S":20,"N":870}'
```

En este ejemplo podemos ver que la primera petición contiene datos de todo (aunque no todas son necesarias porque pueden calcularse unas a partir de las otras como se indica en la documentación de la API) con una edad de 5 años en ese caso concreto. La segunda petición (y sucesivas) tiene valor “t” de inicio igual a 1 y no se indica valor de “H”, porque no es necesario al leer el valor de “S” de la primera simulación, ni valor de “G” porque se calcula en base al valor de “N” que será el indicado por el usuario (o leído del modelo silvícola) para la replantación de la masa.

El script “Solicitudes.py” realiza todos los procesos necesarios para hacer las llamadas necesarias en base al marco temporal, la especie, los datos iniciales del rodal y la silvicultura programada.

2.2 Módulo de suelo

El módulo de suelo parte de una lista que contiene, para cada especie, la proporción de los componentes lignocelulósicos según el tipo de materia orgánica de que se trate (hojarasca o “no maderera, materiales finos y materiales gruesos). El siguiente código es el que genera la lista:

```
# Lista de parámetros para todas las especies
species = {
    "Pp": {"turnover": [0.36, 0.03, 0.073], "nwl": [0.27, 0.51, 0.22], "fwl": [0.03, 0.65, 0.32], "cwl": [0.03, 0.69, 0.28]},
    "Pr": {"turnover": [0.36, 0.03, 0.073], "nwl": [0.27, 0.51, 0.22], "fwl": [0.03, 0.65, 0.32], "cwl": [0.03, 0.69, 0.28]},
    "Ps": {"turnover": [0.50, 0.03, 0.147], "nwl": [0.27, 0.51, 0.22], "fwl": [0.03, 0.65, 0.32], "cwl": [0.03, 0.69, 0.28]},
    "Eg": {"turnover": [0.324, 0.03, 0.047], "nwl": [0.27, 0.51, 0.22], "fwl": [0.03, 0.65, 0.32], "cwl": [0.03, 0.69, 0.28]},
    "Qr": {"turnover": [1, 0.02, 0.03], "nwl": [0.38, 0.36, 0.26], "fwl": [0.03, 0.65, 0.32], "cwl": [0.03, 0.75, 0.22]},
    "Bp": {"turnover": [1, 0.02, 0.03], "nwl": [0.38, 0.36, 0.26], "fwl": [0.03, 0.65, 0.32], "cwl": [0.03, 0.75, 0.22]}
}
nombres_turnover = ["hojas", "ramas", "raíces"]
nombres_litter = ["solubles", "celulosas", "ligninas"]

for especie, parametros in species.items():
    for nombre, vector in parametros.items():
        if nombre == "turnover":
            nombres = nombres_turnover
        else:
            nombres = nombres_litter
        parametros[nombre] = dict(zip(nombres, vector))
```

Siendo: *Turnover*, el ratio de reciclaje de cada parte de los árboles (por ejemplo en el roble “Qr”, el valor es igual a 1 para las hojas, lo que significa que anualmente todas las hojas se reemplazan anualmente, como es lógico al ser una especie de hoja caduca); *nwl*, define el tipo de materia orgánica correspondiente a los elementos no madereros u hojarasca; *fwl*, materia orgánica correspondiente a elementos finos como ramas y raíces finas; *cwl*, materia orgánica correspondiente a elementos gruesos como troncos o raíces gruesas.

A continuación, el script define una variable “Parámetro” que depende de la especie indicada por el usuario en base al código ifn. Así, en base a “Parámetro” se selecciona una u otra parte de la lista anterior.

Con los datos de las coordenadas X-Y del rodal será posible leer los datos de las tres variables climáticas necesarias de los raster de Geoserver siguientes (en el script se pone solamente un valor de ejemplo, faltaría hacer el enlace con geoserver desde Bikenta para obtener el valor):

- Variable "p": 45.tif
- Variable "SumTemp": 46.tif
- Variable "ETP": 47.tif

Con estas variables será posible calcular los valores de unos coeficientes, correspondientes a las ratios de descomposición, que se calculan tal como se puede observar en el script. Además, se selecciona la concentración de cada componente en cada tipo de hojarasca según el valor de la variable "Parametro". A continuación, se transcribe el script de esta parte:

```
# Cálculo parámetros climáticos
p = 500 # Precipitación del período vegetativo. Se lee de Geoserver del raster 45.tif
SumTemp = 5511 # Suma de temperaturas anual. Se lee de Geoserver del raster 46.tif.
ETP = 656.98 # ETP del período vegetativo. Se lee de Geoserver del raster 47.tif.

# Ratios de descomposición
if p > ETP:
    p = ETP
s = 0.6
anwl = min(1, 1 * (1 + 1 * 0.000387 * (SumTemp - 1903) + 0.00325 * ((p - ETP) + 32)))
afwl = min(1, 0.54 * (1 + 1 * 0.000387 * (SumTemp - 1903) + 0.00325 * ((p - ETP) + 32)))
acwl = min(1, 0.0535 * (1 + 1 * 0.000387 * (SumTemp - 1903) + 0.00325 * ((p - ETP) + 32)))
ksol = min(1, np.where(Parametro in range(1, 5), 0.48, 0.82) * (1 + 1 * 0.000387 * (SumTemp - 1903) + 0.00325 * ((p - ETP) + 32)))
kcel = min(1, 0.3 * (1 + 1 * 0.000387 * (SumTemp - 1903) + 0.00325 * ((p - ETP) + 32)))
klig = min(1, 0.22 * (1 + s * 0.000387 * (SumTemp - 1903) + 0.00325 * ((p - ETP) + 32)))
khum1 = min(1, 0.012 * (1 + s * 0.000387 * (SumTemp - 1903) + 0.00325 * ((p - ETP) + 32)))
khum2 = min(1, 0.0012 * (1 + s * 0.000387 * (SumTemp - 1903) + 0.00325 * ((p - ETP) + 32)))

# Concentración de cada componente en cada tipo de hojarasca
Cnwlsol = np.where(Parametro in range(1, 5), 0.27, 0.38)
Cfwlsol = np.where(Parametro in range(1, 5), 0.03, 0.03)
Ccwlsol = np.where(Parametro in range(1, 5), 0.03, 0.03)
Cnwlccl = np.where(Parametro in range(1, 5), 0.51, 0.36)
Cfwlccl = np.where(Parametro in range(1, 5), 0.65, 0.65)
Ccwlccl = np.where(Parametro in range(1, 5), 0.69, 0.75)
Cnwllig = np.where(Parametro in range(1, 5), 0.22, 0.26)
Cfwllig = np.where(Parametro in range(1, 5), 0.32, 0.32)
Ccwllig = np.where(Parametro in range(1, 5), 0.28, 0.22)
```

La parte final del módulo consiste en calcular, en primer lugar, la cantidad de carbono anual en los diferentes tipos de materia orgánica del suelo (*nwl*, *fwl*, *cwl*) y, posteriormente, la cantidad de carbono almacenada componentes de la materia orgánica del suelo más complejos (elementos solubles "Sol", celulosas "Cel", ligninas "Lig", humus simple "Hum1" y humus complejo "Hum2"). Para realizar estos cálculos se parte de los datos de CO₂ almacenados en la masa forestal en pie, provenientes de la API de la universidad, los cuales es preciso transformar a valores de carbono empleando un factor de conversión a carbono, denominado en el código como "fC", el cual es igual a 12/44. El código en el que se definen con detalle estos cálculos se muestra a continuación:

```
fC=12/44 # Factor de conversión de CO2 a Carbono.
```

```
# Cálculos de las fracciones en el suelo
datos['nwl'] = 0.0# Inicializamos el DataFrame con los valores iniciales de nwl, fwl y cwl
datos['fwl'] = 0.0# Inicializamos el DataFrame con los valores iniciales de nwl, fwl y cwl
datos['cwl'] = 0.0# Inicializamos el DataFrame con los valores iniciales de nwl, fwl y cwl
# Iterar sobre las filas del DataFrame empezando desde la segunda fila
for i in range(1, len(datos)):
    try:
        nwl_calculation = (
            datos.at[i-1, 'nwl'] - datos.at[i-1, 'nwl'] * anwl +
            fC * datos.at[i-1, 'CO2p_follaje'] * Imput.get('turnover', {}).get('hojas', 0) +
            fC * datos.at[i-1, 'CO2_raices'] * Imput.get('turnover', {}).get('raices', 0) * (
                0 if fC * datos.at[i-1, 'CO2p_follaje'] * Imput.get('turnover', {}).get('hojas', 0) + fC * datos.at[i-1,
'CO2p_ramas'] * Imput.get('turnover', {}).get('ramas', 0) == 0 else
                (fC * datos.at[i-1, 'CO2p_follaje'] * Imput.get('turnover', {}).get('hojas', 0)) /
                (fC * datos.at[i-1, 'CO2p_follaje'] * Imput.get('turnover', {}).get('hojas', 0) + fC * datos.at[i-1,
'CO2p_ramas'] * Imput.get('turnover', {}).get('ramas', 0))
            ) +
            fC * datos.at[i, 'CO2e_follaje'] * datos.at[i, 'restos_suelo']
        )
        datos.at[i, 'nwl'] = nwl_calculation

        fwl_calculation = (
            datos.at[i-1, 'fwl'] - datos.at[i-1, 'fwl'] * afwl +
            fC * datos.at[i-1, 'CO2p_ramas'] * Imput.get('turnover', {}).get('ramas', 0) +
            fC * datos.at[i-1, 'CO2_raices'] * Imput.get('turnover', {}).get('raices', 0) * (
                0 if fC * datos.at[i-1, 'CO2p_follaje'] * Imput.get('turnover', {}).get('hojas', 0) + fC * datos.at[i-1,
'CO2p_ramas'] * Imput.get('turnover', {}).get('ramas', 0) == 0 else
                (fC * datos.at[i-1, 'CO2p_ramas'] * Imput.get('turnover', {}).get('ramas', 0)) /
                (fC * datos.at[i-1, 'CO2p_follaje'] * Imput.get('turnover', {}).get('hojas', 0) + fC * datos.at[i-1,
'CO2p_ramas'] * Imput.get('turnover', {}).get('ramas', 0))
            ) +
            fC * datos.at[i, 'CO2e_ramas'] * datos.at[i, 'restos_suelo']
        )
        datos.at[i, 'fwl'] = fwl_calculation

        cwl_calculation = (
            datos.at[i-1, 'cwl'] - datos.at[i-1, 'cwl'] * acwl +
            fC * datos.at[i, 'CO2e_troncos'] * (1 - datos.at[i, 'Ve_hasta_dmin3'] / datos.at[i, 'Ve']) * datos.at[i,
'restos_suelo']
            if datos.at[i, 'Ve_hasta_dmin3'] > 0 and datos.at[i, 'Ve'] > 0 else
            datos.at[i-1, 'cwl'] - datos.at[i-1, 'cwl'] * acwl
        )
        datos.at[i, 'cwl'] = cwl_calculation

    except Exception as e:
        print(f"Error en la fila {i}: {e}")

# Cálculos de carbono almacenado en los componentes de la materia orgánica del suelo más complejos
datos["Sol"] = 0.0
datos["Cel"] = 0.0
datos["Lig"] = 0.0
datos["Hum1"] = 0.0
datos["Hum2"] = 0.0
datos["CT_Soil"] = 0.0
```

```

for i in range(1, len(datos)):
    try:
        datos.at[i, 'Sol'] = (datos.at[i-1, 'nwl'] * anwl * Cnwlsol +
                             datos.at[i-1, 'fwl'] * afwl * Cfwlsol +
                             datos.at[i-1, 'cwl'] * acwl * Ccwlslol -
                             ksol * datos.at[i-1, 'Sol'] +
                             datos.at[i-1, 'Sol'])

        datos.at[i, 'Cel'] = (datos.at[i-1, 'nwl'] * anwl * Cnwlccl +
                             datos.at[i-1, 'fwl'] * afwl * Cfwlccl +
                             datos.at[i-1, 'cwl'] * acwl * Ccwlcccl -
                             kcel * datos.at[i-1, 'Cel'] +
                             datos.at[i-1, 'Cel'])

        datos.at[i, 'Lig'] = (datos.at[i-1, 'nwl'] * anwl * Cnwllig +
                             datos.at[i-1, 'fwl'] * afwl * Cfwllig +
                             datos.at[i-1, 'cwl'] * acwl * Ccwlilig -
                             klig * datos.at[i-1, 'Lig'] +
                             datos.at[i-1, 'Lig'] +
                             0.2 * ksol * datos.at[i-1, 'Sol'] +
                             0.2 * kcel * datos.at[i-1, 'Cel'] -
                             0.2 * (datos.at[i-1, 'Lig'] - datos.at[i-1, 'Lig'] * klig))

        datos.at[i, 'Hum1'] = (0.2 * klig * datos.at[i-1, 'Lig'] -
                               datos.at[i-1, 'Hum1'] * khum1 +
                               datos.at[i-1, 'Hum1'])

        datos.at[i, 'Hum2'] = (0.2 * khum1 * datos.at[i-1, 'Hum1'] -
                               khum2 * datos.at[i-1, 'Hum2'] +
                               datos.at[i-1, 'Hum2'])

        datos.at[i, 'CT_Soil'] = (datos.at[i, 'nwl'] +
                                  datos.at[i, 'fwl'] +
                                  datos.at[i, 'cwl'] +
                                  datos.at[i, 'Sol'] +
                                  datos.at[i, 'Cel'] +
                                  datos.at[i, 'Lig'] +
                                  datos.at[i, 'Hum1'] +
                                  datos.at[i, 'Hum2'])
    except Exception as e:
        print(f"Error en la fila {i}: {e}")

Resultados_Suelo = datos[["Año", "nwl", "fwl", "cwl", "Sol", "Cel", "Lig", "Hum1", "Hum2", "CT_Soil"]] #
Exporta los resultados del suelo para integrar con el resto de módulos

```

“CT_Soil” contiene el total acumulado en el suelo hasta ese año.

2.3 Módulo de productos

El módulo productos necesita saber inicialmente cuál es el objetivo productivo del monte, de modo que para ello se genera la variable “Objetivo” que toma el valor a partir de lo indicado en el formulario por el usuario. Las opciones son: Estructural, Sierra, Tablero y Papel.

En función de dicha variable se calcularán los stocks de carbono en las diferentes tipologías de longevidad:

- Estr: madera que queda almacenada en productos de uso estructural (la más longeva).
- LP: madera que queda almacenada en productos durante un largo plazo.
- MP: madera que queda almacenada en productos durante un plazo medio.
- CP: madera que queda almacenada en productos durante un corto plazo.

También es necesario que Bikenta haya generado una columna llamada “Actuacion” a partir de la información silvícola, de modo que las claras tomen valor “CI” y las cortas finales tomen el valor “CF”, ya que es necesaria esta información como se puede ver en el código del script:

Establezco el objetivo leído de la interfaz

Objetivo = "Estructural" # Se lee del formulario donde se indica el objetivo de la gestión.

Calculo la adición anual a cada tipología de longevidad-

if Objetivo == "Estructural":

```
datos['U_Estr'] = fC * datos['CO2e_troncos'] * (datos['Ve_hasta_dmin3'] / datos['Ve']) * (
    datos['Ve1'] / datos['Ve_hasta_dmin3']) * np.where(datos['Actuacion'] == "CF", 0.5, 0)
```

```
datos['U_LP'] = fC * datos['CO2e_troncos'] * (datos['Ve_hasta_dmin3'] / datos['Ve']) * (
    datos['Ve1'] / datos['Ve_hasta_dmin3']) * np.where(datos['Actuacion'] == "CF", 0,
    np.where(datos['Actuacion'] == "CI", 0.5, 0))
```

```
datos['U_MP'] = (fC * datos['CO2e_troncos'] * (datos['Ve_hasta_dmin3'] / datos['Ve']) * (
    datos['Ve1'] / datos['Ve_hasta_dmin3']) * 0.4 +
    fC * datos['CO2e_troncos'] * (datos['Ve_hasta_dmin3'] / datos['Ve']) * (
    datos['Ve2'] / datos['Ve_hasta_dmin3']) * 0.9 +
    fC * datos['CO2e_troncos'] * (datos['Ve_hasta_dmin3'] / datos['Ve']) * (
    datos['Ve3'] / datos['Ve_hasta_dmin3']) * 0.9)
```

```
datos['U_CP'] = 0
```

```
datos['U_Bio'] = (fC * datos['CO2e_troncos'] * (1 - datos['Ve_hasta_dmin3'] / datos['Ve']) * (
    1 - datos['restos_suelo']) +
    fC * datos['CO2e_ramas'] * (1 - datos['restos_suelo']) +
    fC * datos['CO2e_follaje'] * (1 - datos['restos_suelo']) +
    fC * datos['CO2e_troncos'] * (datos['Ve_hasta_dmin3'] / datos['Ve']) * (
    datos['Ve1'] / datos['Ve_hasta_dmin3']) * 0.1 +
    fC * datos['CO2e_troncos'] * (datos['Ve_hasta_dmin3'] / datos['Ve']) * (
    datos['Ve2'] / datos['Ve_hasta_dmin3']) * 0.1 +
    fC * datos['CO2e_troncos'] * (datos['Ve_hasta_dmin3'] / datos['Ve']) * (
    datos['Ve3'] / datos['Ve_hasta_dmin3']) * 0.1)
```

elif Objetivo == "Sierra":

```
datos['U_Estr'] = fC * datos['CO2e_troncos'] * (datos['Ve_hasta_dmin3'] / datos['Ve']) * (
    datos['Ve1'] / datos['Ve_hasta_dmin3']) * np.where(datos['Actuacion'] == "CF", 0.25, 0)
```

```
datos['U_LP'] = fC * datos['CO2e_troncos'] * (datos['Ve_hasta_dmin3'] / datos['Ve']) * (
    datos['Ve1'] / datos['Ve_hasta_dmin3']) * np.where(datos['Actuacion'] == "CF", 0.25,
    np.where(datos['Actuacion'] == "CI", 0.5, 0))
```

```
datos['U_MP'] = (fC * datos['CO2e_troncos'] * (datos['Ve_hasta_dmin3'] / datos['Ve']) * (
    datos['Ve1'] / datos['Ve_hasta_dmin3']) * 0.4 +
    fC * datos['CO2e_troncos'] * (datos['Ve_hasta_dmin3'] / datos['Ve']) * (
    datos['Ve2'] / datos['Ve_hasta_dmin3']) * 0.9 +
    fC * datos['CO2e_troncos'] * (datos['Ve_hasta_dmin3'] / datos['Ve']) * (
    datos['Ve3'] / datos['Ve_hasta_dmin3']) * 0.9)
```

```
datos['U_CP'] = 0
```

```
datos['U_Bio'] = (fC * datos['CO2e_troncos'] * (1 - datos['Ve_hasta_dmin3'] / datos['Ve']) * (
    1 - datos['restos_suelo']) +
    fC * datos['CO2e_ramas'] * (1 - datos['restos_suelo']) +
    fC * datos['CO2e_follaje'] * (1 - datos['restos_suelo']) +
```

```

        fC * datos['CO2e_troncos'] * (datos['Ve_hasta_dmin3'] / datos['Ve']) * (
            datos['Ve1'] / datos['Ve_hasta_dmin3']) * 0.1 +
        fC * datos['CO2e_troncos'] * (datos['Ve_hasta_dmin3'] / datos['Ve']) * (
            datos['Ve2'] / datos['Ve_hasta_dmin3']) * 0.1 +
        fC * datos['CO2e_troncos'] * (datos['Ve_hasta_dmin3'] / datos['Ve']) * (
            datos['Ve3'] / datos['Ve_hasta_dmin3']) * 0.1)
elif Objetivo == "Tablero":
    datos['U_Estr'] = 0
    datos['U_LP'] = 0
    datos['U_MP'] = (fC * datos['CO2e_troncos'] * (datos['Ve_hasta_dmin3'] / datos['Ve']) *
        (datos['Ve1'] / datos['Ve_hasta_dmin3']) * 0.9 +
        fC * datos['CO2e_troncos'] * (datos['Ve_hasta_dmin3'] / datos['Ve']) *
        (datos['Ve2'] / datos['Ve_hasta_dmin3']) * 0.9 +
        fC * datos['CO2e_troncos'] * (datos['Ve_hasta_dmin3'] / datos['Ve']) *
        (datos['Ve3'] / datos['Ve_hasta_dmin3']) * 0.9)
    datos['U_CP'] = 0
    datos['U_Bio'] = (fC * datos['CO2e_troncos'] * (1 - datos['Ve_hasta_dmin3'] / datos['Ve']) *
        (1 - datos['restos_suelo']) +
        fC * datos['CO2e_ramas'] * (1 - datos['restos_suelo']) +
        fC * datos['CO2e_follaje'] * (1 - datos['restos_suelo']) +
        fC * datos['CO2e_troncos'] * (datos['Ve_hasta_dmin3'] / datos['Ve']) *
        (datos['Ve1'] / datos['Ve_hasta_dmin3']) * 0.1 +
        fC * datos['CO2e_troncos'] * (datos['Ve_hasta_dmin3'] / datos['Ve']) *
        (datos['Ve2'] / datos['Ve_hasta_dmin3']) * 0.1 +
        fC * datos['CO2e_troncos'] * (datos['Ve_hasta_dmin3'] / datos['Ve']) *
        (datos['Ve3'] / datos['Ve_hasta_dmin3']) * 0.1)
elif Objetivo == "Papel":
    datos['U_Estr'] = 0
    datos['U_LP'] = 0
    datos['U_MP'] = 0
    datos['U_CP'] = (fC * datos['CO2e_troncos'] * (datos['Ve_hasta_dmin3'] / datos['Ve']) *
        (datos['Ve1'] / datos['Ve_hasta_dmin3']) * 0.9 +
        fC * datos['CO2e_troncos'] * (datos['Ve_hasta_dmin3'] / datos['Ve']) *
        (datos['Ve2'] / datos['Ve_hasta_dmin3']) * 0.9 +
        fC * datos['CO2e_troncos'] * (datos['Ve_hasta_dmin3'] / datos['Ve']) *
        (datos['Ve3'] / datos['Ve_hasta_dmin3']) * 0.9)
    datos['U_Bio'] = (fC * datos['CO2e_troncos'] * (1 - datos['Ve_hasta_dmin3'] / datos['Ve']) *
        (1 - datos['restos_suelo']) +
        fC * datos['CO2e_ramas'] * (1 - datos['restos_suelo']) +
        fC * datos['CO2e_follaje'] * (1 - datos['restos_suelo']) +
        fC * datos['CO2e_troncos'] * (datos['Ve_hasta_dmin3'] / datos['Ve']) *
        (datos['Ve1'] / datos['Ve_hasta_dmin3']) * 0.1 +
        fC * datos['CO2e_troncos'] * (datos['Ve_hasta_dmin3'] / datos['Ve']) *
        (datos['Ve2'] / datos['Ve_hasta_dmin3']) * 0.1 +
        fC * datos['CO2e_troncos'] * (datos['Ve_hasta_dmin3'] / datos['Ve']) *
        (datos['Ve3'] / datos['Ve_hasta_dmin3']) * 0.1)
datos.fillna(0, inplace=True)

```

A continuación, se definen las vidas útiles (HL) de cada tipo de producto. Para calcular los stocks en cada caso se emplean de nuevo los datos recibidos de la API de la universidad, la cual ya tuvo en cuenta los valores de los diámetros en punta delgada de cada destino en base a la tabla que los define para cada especie y objetivo. Aquí se indica el código necesario para realizar los cálculos:

```
# Cálculo de la pérdida de carbono almacenado en productos en base a su vida útil -----
# 1º Establecemos vida útil e inicializamos las columnas con valores iguales a 0
HL = {"Estructural": 90, "Sierra": 35, "Tablero": 25, "Papel": 2, "Bioenergía": 0}
datos["Estr"] = 0.0
datos["LP"] = 0.0
datos["MP"] = 0.0
datos["CP"] = 0.0
datos["Bio"] = 0.0
for i in range(1, len(datos)):
    datos.at[i, "Estr"] = datos.at[i-1, "Estr"] * np.exp(-np.log(2) / HL["Estructural"]) + datos.at[i, "U_Estr"]
    datos.at[i, "LP"] = datos.at[i-1, "LP"] * np.exp(-np.log(2) / HL["Sierra"]) + datos.at[i, "U_LP"]
    datos.at[i, "MP"] = datos.at[i-1, "MP"] * np.exp(-np.log(2) / HL["Tablero"]) + datos.at[i, "U_MP"]
    datos.at[i, "CP"] = datos.at[i-1, "CP"] * np.exp(-np.log(2) / HL["Papel"]) + datos.at[i, "U_CP"]
    datos.at[i, "Bio"] = datos.at[i-1, "Bio"] * np.exp(-np.log(2) / HL["Bioenergía"]) + datos.at[i, "U_Bio"]
datos["Total_Productos"] = datos["Estr"] + datos["LP"] + datos["MP"] + datos["CP"] + datos["Bio"]
Resultados_Productos = datos[["Año", "Estr", "LP", "MP", "CP", "Bio", "Total_Productos"]]
```

“Total_Productos” contiene el valor total acumulado en productos hasta ese año.

2.4 Módulo de bioenergía

El módulo de bioenergía calcula el CO₂ dejado de consumir por el uso de restos forestales en lugar de combustibles fósiles. Se parte de los datos de biomasa (nótese que este módulo hace los cálculos a partir de valores de biomasa en vez de valores de CO₂ como en los módulos anteriores), devueltos por la API de la universidad para, a partir del valor del porcentaje de restos dejados en el suelo, que se marca en el plan silvícola (ya sea cogido del modelo silvícola tabulado en la línea base o el indicado por el usuario en el formulario).

Así, por un lado, se calcula la biomasa disponible para su uso como bioenergía proveniente de los restos de corta y, por otra parte, la biomasa disponible para su uso como bioenergía proveniente de los restos desperdiciados en la industria:

```
# Calculo la biomasa de cortas en campo y la biomasa reutilizada en la industria
datos["W_restos_corta"] = 0.0
datos["W_restos_industria"] = 0.0

for i in range(1, len(datos)): # Empezar desde el segundo elemento (i=1) para evitar el caso i=0
    # Obtener los valores previos
    We_troncos = datos.at[i-1, "We_troncos"]
    Ve_hasta_dmin3 = datos.at[i-1, "Ve_hasta_dmin3"]
    Ve = datos.at[i-1, "Ve"]
    restos_suelo = datos.at[i-1, "restos_suelo"]
    We_follaje = datos.at[i-1, "We_follaje"]
    We_ramas = datos.at[i-1, "We_ramas"]
    # Calcular W_restos_corta
    if Ve != 0: # Evitar división por cero
        datos.at[i, "W_restos_corta"] = (
            We_troncos * (1 - Ve_hasta_dmin3 / Ve) * restos_suelo +
            We_follaje * restos_suelo +
            We_ramas * restos_suelo
        )
    else:
        datos.at[i, "W_restos_corta"] = 0
    # Calcular W_restos_industria
```

```

if Ve != 0: # Evitar división por cero
    datos.at[i, "W_restos_industria"] = (
        We_troncos * (Ve_hasta_dmin3 / Ve) * 0.1
    )
else:
    datos.at[i, "W_restos_industria"] = 0
# Calcular W_restos
datos.at[i, "W_restos"] = datos.at[i, "W_restos_corta"] + datos.at[i, "W_restos_industria"]
datos.fillna(0, inplace=True)

```

A continuación, se establecen los factores relevantes en las emisiones de CO₂ para los combustibles fósiles y para la biomasa en base a valores dados por el Ministerio para la Transición Ecológica. Estos factores son:

- Factor de Emisión (FE, $kgCO_2e/ud$).
- Eficiencia de la tecnología de combustión (Efi, %). Se establece el mismo valor para ambos casos pues se supone que se usa la misma tecnología y la diferencia está en el tipo de combustible y no en la tecnología empleada.
- Calor Específico (Heat, MJ/kg) del material.

Ya, por último, se calcula el CO_{2e} emitido por el uso de biomasa (CO_{2e_alt}) y el que se hubiera emitido por el uso de combustibles fósiles (CO_{2e_subs}) para producir la misma cantidad de energía. De este modo, calculando la diferencia de emisiones entre uno y otro se obtiene el CO₂ mitigado por el uso de biomasa y, empleando el ya mencionado factor de conversión (fC) se obtiene el valor del carbono almacenado (mitigado) por el uso de biomasa:

```

# Lista emisiones tecnología substituida y alternativa-----
Substituida = {
    "FE": 2.721,
    "Efi": 25,
    "Heat": 43.33
}
Alternativa = {
    "FE": 0.137,
    "Efi": 25,
    "Heat": 15
}
datos["CO2e_subs"] = 0.0
datos["CO2e_alt"] = 0.0
datos["GHGe"] = 0.0
for i in range(1, len(datos)):
    datos.at[i, "CO2e_subs"] = datos.at[i, "W_restos"] * Substituida["FE"] * Alternativa["Heat"] /
    Substituida["Heat"]
    datos.at[i, "CO2e_alt"] = datos.at[i, "W_restos"] * Alternativa["FE"] * Substituida["Heat"] /
    Alternativa["Heat"]
    if i == 1:
        datos.at[i, "GHGe"] = datos.at[i, "CO2e_subs"] - datos.at[i, "CO2e_alt"]
    else:
        datos.at[i, "GHGe"] = datos.at[i, "CO2e_subs"] - datos.at[i, "CO2e_alt"] + datos.at[i-1, "GHGe"]
    datos.at[i, "C_bioenergia"] = fC * datos.at[i, "GHGe"]
datos.fillna(0, inplace=True)
Resultados_bioenergia = datos[["Año", "GHGe", "C_bioenergia"]]

```

2.5 Módulo de incendios

En este módulo se parte de una lista en la que, para cada especie, se establecen los valores de los factores de combustión (en base al índice de peligrosidad) y los factores de emisión. Así, lo primero que se hace es crear una función para seleccionar los factores, en la cual se define la lógica según la cual la peligrosidad es muy alta, alta, media o baja. Este hecho dependerá de la existencia de podas y/o desbroces a lo largo del plan de actuaciones de mejora establecidos. Por defecto (si no hay podas ni desbroces) la peligrosidad es alta y se asigna valor 0 a las variables “poda” y “desbroce”; si hay una de las dos actuaciones desciende a media y se asigna el valor 1 para la variable que corresponda a la actuación programada; en caso de que haya programadas actuaciones de ambos tipos, la peligrosidad es baja y se asigna valor 1 tanto a “poda” como a “desbroce”. El siguiente script muestra dicha función (faltaría la programación para completar el valor “poda” o “desbroce” con 1 o 0 en función de la información del plan de mejoras):

```
# Definición de la especie: el valor sp se lee del formulario.
if sp == 26:
    Especie = "Pp"
elif sp == 28:
    Especie = "Pr"
elif sp == 21:
    Especie = "Ps"
elif sp == 61:
    Especie = "Eg"
elif sp == 41:
    Especie = "Qr"
elif sp == 73:
    Especie = "Bp"

# Función de selección de los factores con la Lista de factores de combustión y factores de emisión en
función de la especie
def obtener_Fcomb_FE(Especie, poda, desbroce):
    Factores = {
        "Pp": {
            "Fcomb": {"Muy Alta": 0.65, "Alta": 0.56, "Media": 0.47, "Baja": 0.25},
            "FEM": 1.627
        },
        "Pr": {
            "Fcomb": {"Muy Alta": 0.65, "Alta": 0.56, "Media": 0.47, "Baja": 0.25},
            "FEM": 1.627
        },
        "Ps": {
            "Fcomb": {"Muy Alta": 0.65, "Alta": 0.56, "Media": 0.47, "Baja": 0.25},
            "FEM": 1.627
        },
        "Eg": {
            "Fcomb": {"Muy Alta": 0.65, "Alta": 0.56, "Media": 0.47, "Baja": 0.25},
            "FEM": 1.627
        },
        "Qr": {
            "Fcomb": {"Muy Alta": 0.56, "Alta": 0.48, "Media": 0.40, "Baja": 0.25},
            "FEM": 1.393
        },
    }
```

```
"Bp": {
  "Fcomb": {"Muy Alta": 0.56, "Alta": 0.48, "Media": 0.40, "Baja": 0.25},
  "FEM": 1.393
}
}
```

```
Fcomb = None
if poda == 0 and desbroce == 0:
  Fcomb = "Alto"
elif poda == 1 and desbroce == 0:
  Fcomb = "Media"
elif poda == 0 and desbroce == 1:
  Fcomb = "Media"
elif poda == 1 and desbroce == 1:
  Fcomb = "Bajo"
else:
  return "Valores de poda o desbroce no válidos"

FEM = Factores[Especie]["FEM"]
Fcomb_valor = Factores[Especie]["Fcomb"][Fcomb]

return {"Fcomb": Fcomb_valor, "FEM": FEM}
```

Además, es preciso conocer la probabilidad de incendio, la cual se lee del ráster de Geoserver "81.tif". Con todos los datos, ya se puede calcular el valor del CO₂ (y del Carbono) que potencialmente se perdería por culpa de los incendios forestales, el cual toma valor negativo y será más o menos negativo en función de la existencia o no de actuaciones de mejora previstas:

```
Probabilidad=0.0649#Lectura del raster 81.tif
poda = 1 #Cuando es positivo (hay podas programados) toma valor 1
desbroce = 0 #Cuando es negativo (no hay desbroces programados) toma valor 0
resultados = obtener_Fcomb_FE(Especie, poda, desbroce)
Fcomb=(resultados["Fcomb"])
Fem=(resultados["FEM"])

datos["C_incendios"] = 0.0 # Inicializo el valor

# Calcular los valores de C_incendios
for i in range(len(datos)):
  if i == 0:
    datos.at[i, "C_incendios"] = -datos.at[i, "Wp_aerea"] * Fcomb * Fem * (Probabilidad / 100) * fC
  else:
    datos.at[i, "C_incendios"] = -datos.at[i, "Wp_aerea"] * Fcomb * Fem * (Probabilidad / 100) * fC +
    datos.at[i - 1, "C_incendios"]
# Seleccionar las columnas "Año" y "C_incendios" de los resultados
Resultados_incendios = datos[["Año", "C_incendios"]]
```

METODOLOGÍA DE ABSORCIÓN DE CO₂

1. Gestores forestales

Cálculo CO₂ absorbido = CO₂ crecimiento – mortalidad – riesgo de incendios-emisiones tratamientos

a) Evaluación Ex ante

Según sean proyectos de repoblación o proyectos de manejo de masas existentes:

a) Repoblación/Forestación:

- I. LINEA BASE: evolución hasta corta final según modelos dinámicos de crecimiento para una calidad de estación media, salvo casos en que previo a la corta final de la misma especie se disponga de inventario forestal con datos de edad y altura dominante, en cuyo caso la simulación se hace para la calidad de estación así calculada. Transcurridos los años suficientes para evaluar la calidad de estación (10 años en pinos, 7 en eucaliptos y 20 en roble), se hace la primera comprobación y, en ese momento se determina la calidad de estación para, posteriormente simular los restantes años hasta la corta final (turno tecnológico: crecimiento medio = crecimiento corriente) en base a la calidad de estación obtenida. En estas mediciones posteriores se extraerá una barrena de un árbol medio de la parcela para estimar la densidad básica y así, multiplicando por el valor de volumen total, obtener el valor de biomasa del fuste (el valor de biomasa de hojas, ramas y raíces será el de los modelos de biomasa), el cual se multiplica por el contenido en carbono (%) para calcular el carbono absorbido en cada instante. De este modo, las proyecciones de carbono acumulado se basan en la proyección de la biomasa estimada esperada, multiplicando la biomasa del fuste por este valor de densidad, en lugar de aplicar directamente el modelo de estimación de biomasa. Este valor de densidad se irá actualizando en las sucesivas evaluaciones del proyecto. (En un futuro incluso podría ser posible elaborar modelos de predicción de la densidad en base a la edad y variables de masa con la misma metodología de creación de curvas de calidad de estación con lo que anualmente se actualizaría este valor de densidad de la proyección).
- II. LINEA DE MANEJO SILVÍCOLA: se parte de una calidad de estación media (salvo que haya datos de la masa previa y ésta sea de la misma especie) y se planifica un calendario de actuaciones en base a criterios técnicos, el cual se emplea para la simulación mediante los modelos dinámicos del libro *Herramientas silvícolas para la gestión forestal sostenible en Galicia*". Igualmente, en el caso del fuste, se calcula la densidad de la madera y se estima la biomasa a partir de este dato y de la proyección de volumen del fuste.
- III. Crecimiento y mortalidad: se emplearán los modelos de herramientas silvícolas para cada especie que simulen los crecimientos y la mortalidad, tanto para línea base como para línea de gestión.

- IV. En el momento de una actuación que suponga la remoción de pies mayores (diámetro mayor de 7 cm), se asume que la biomasa extraída será empleada en productos sustitutorios de otros provenientes de fuentes de energía no renovables, con lo que no restará en el stock de carbono almacenado por los gestores forestales. Sin embargo, sí se contabilizarán las emisiones de CO₂ causadas por la maquinaria (desbrozadoras, motosierra...etc.) las cuales se parametrizarán en base a la metodología propuesta por “Castella”.
- V. Se estima el riesgo de incendios en base a la fórmula del IPCC multiplicando por una probabilidad de incendio como propone “Castella”. Esta probabilidad se calcula en base a los datos de incendios forestales de los últimos años en el distrito forestal donde se halle el monte, con lo que se obtiene una media anual de hectáreas calcinadas por distrito, dividida por el total de hectáreas de monte (se considera el forestal arbolado y el matorral). Actualmente, no se dispone de estadísticas forestales demasiado fiables para esta superficie, pero puede usarse el último IFN o la norma de los 2/3 (2/3 de Galicia son monte de los cuales 2/3 son arbolados) para estimar la superficie forestal de cada distrito de índole forestal. En cuanto el IFCG concluya sus resultados y publique los mapas se obtendrá un valor mucho más preciso y se usará dicha fuente de información. Además, si es posible obtener estadísticas de la Xunta por municipios en lugar de por Distritos, la probabilidad de ocurrencia será calculada a menor escala. Dicha probabilidad tendrá una componente social, lo cual tiene sentido porque se estima que el 90% de los incendios son provocados. Los otros parámetros de la ecuación son los factores de emisión y los factores de combustión. Para los factores de combustión, “Castella” propone cuatro clases de voracidad del fuego, para las que estima sus correspondientes factores, en función de si se trata de coníferas o de frondosas, los cuales serán empleados en esta metodología (asumiendo para eucalipto los factores de del pino dado que, en materia de propagación de incendios, no se comporta como una frondosa caducifolia común). Se asume una intensidad alta en caso de no gestión y se asume la intensidad inmediatamente inferior (intensidad media) en caso de gestión. Los factores de emisión serán los indicados en la misma publicación.

b) Manejo de masas preexistentes

- I. LINEA BASE: se realiza una medición en el momento de inicio del proyecto, tomando datos de inventario forestal tradicional, de edad (mediante el conteo de anillos de una barrena extraída de la base de tres pies mayores) y densidad básica. Así se obtiene la calidad de estación y se proyecta a futuro empleando las ecuaciones de la universidad (con la salvedad de la biomasa del fuste para la que se emplea el resultado de la densidad básica de la barrena extraída).
- II. LINEA DE MANEJO SILVICOLA: partiendo de los datos de medición en campo, se planifica un calendario de actuaciones en base a criterios técnicos, el cual se emplea para la simulación mediante los modelos dinámicos del libro *Herramientas silvícolas para la gestión forestal sostenible en Galicia*. Igualmente, en el caso del fuste, se calcula la densidad de la madera y se estima la biomasa a partir de este dato y de la proyección de volumen del fuste.
- III. El cálculo de emisiones por trabajos silvícolas y riesgo de incendios se lleva a cabo del mismo modo descrito en el apartado A.

Evaluación Ex post

La evaluación ex post se lleva a cabo de igual manera en proyectos de repoblación/forestación que en proyectos de manejo. Consistirá en lo siguiente:

- I. Inventario forestal en campo de las parcelas incluidas en el la masa. Se dejará un bosque de la masa sin gestionar para comparar los resultados de la gestión forestal. Este bosque tendrá un tamaño mínimo de 2000 m² para evitar el efecto borde, pero en masas de menos de 1 ha, se asume el valor estimado por los modelos para obtener el valor sin gestión. Estas parcelas serán geo-posicionadas y marcadas en el terreno para las sucesivas evaluaciones. Estos datos de campo serán los datos de partida para la simulación ex ante siguiente, tanto para línea base (en caso de no ser posible por las dimensiones del bosque hacer parcela consistirá simplemente en seguir la proyección ya establecida) como para la línea de gestión.
- II. Con los datos de inventario se podrá calcular la biomasa real en ambas líneas, empleando los valores de densidad de las barrenas extraídas para el cálculo de la biomasa del fuste. Esto nos dará el valor de biomasa en pie.
- III. A estos valores de biomasa en pie se le aplicará, para el caso de la línea de gestión, las reducciones por trabajos silvícolas y, en caso de haber extraído biomasa en alguna clara intermedia, se le sumarán los valores de biomasa extraída, dado que los gestores son los productores de biomasa y toda esa biomasa extraída existe gracias a que ha sido fijada por ellos dentro de la duración del proyecto de son absorción y son realmente las industrias las que extraen ese producto para mantener su actividad y a los que se les debe restar la biomasa extraída en su balance de carbono.
- IV. No se aplican las reducciones por incendios forestales en caso de que no haya habido un incendio dado que es una evaluación de la realidad en ese momento. En caso de incendio, será necesario inventariar de nuevo las existencias y emplearlas como punto de partida para un nuevo proyecto de absorción.

2. Industrias forestales

$CO_2 = - CO_2 \text{ extraído} - \text{emisiones transporte} - \text{Emisiones proceso productivo} + \text{Mitigación por uso de biomasa en la producción de su energía (o venta de biomasa a otra industria forestal)} + \text{Carbono capturado en los productos en función de su vida útil} = \text{créditos que necesitan comprar para ser neutros.}$

Ex ante

La evaluación ex ante consistirá en lo siguiente:

- I. Se computarán negativamente los valores de biomasa extraída en la masa de los fustes (la de ramas, raíces y hojas ya se les restan a los propietarios), dado que son éstos los que extraen biomasa del medio para mantener su actividad productiva. A este valor se le sumará también el consumo de combustibles en el aprovechamiento y en el transporte. Sin embargo, en aquellos casos en que la biomasa provenga de masas no adscritas a proyectos de absorción de CO₂, ésta no restará ni sumará, siempre y cuando se justifique la existencia de gestión

- (proyectos de ordenación, documentos simples de gestión, adhesión a buenas prácticas silvícolas) o su posterior plantación.
- II. Se computarán negativamente los consumos de su proceso productivo, en base a factores de emisión publicados por el MITECO. Sin embargo, se computarán positivamente, a modo de mitigación, las emisiones reducidas en base al uso de los desechos como biomasa en lugar de energías no renovables o bien la cantidad de biomasa de desechos vendida a otra industria de producción de biomasa.
 - III. Se computa positivamente la cantidad retenida en productos y se va anualmente restando ese stock en base a la vida útil de los productos empleando la ecuación propuesta por el MITECO.
 - IV. LINEAS BASE: en caso de pino, la línea base se fundamenta en que su proceso productivo está orientado a la producción de tableros de partículas y, en caso de eucalipto, está orientado a la producción de pasta para papel. En ambos casos la línea base no considera reciclaje.
 - V. LINEAS DE CALIDAD: en caso de pino y eucalipto un % de la madera se destinará a madera de calidad, de mayor vida útil, y el resto a sus usos habituales de producción de tableros o papel.
 - VI. De lo anteriormente expuesto, se deduce que, aunque la industria compute negativamente el 100% de la biomasa de los fustes extraída, acaba recuperando un gran porcentaje de este consumo en forma de productos y mitigación de emisiones por consumo energético. De todos modos, siempre será un balance negativo, lo cual es lógico por tratarse de industrias extractivas, pero será más o menos negativo en función del destino final de los productos, primando industrias que generen productos de calidad con mayor durabilidad, y su mayor o menor reciclaje. Este valor negativo podrá compensarse acudiendo al mercado de carbono, comprando a los gestores forestales el valor de los créditos de CO₂ emitidos por éstos. Una alternativa también posible es que las industrias acuerden con los propietarios hacerse cargo de la repoblación/plantación de sus terrenos y su gestión, caso en el cual las industrias podrán añadir a su balance de carbono los stocks almacenados en las masas forestales siguiendo la misma metodología descrita para los gestores forestales.
 - VII. REMATANTES: en Galicia tiene gran importancia la figura del rematante. En este caso, la metodología sería la misma, pudiendo compensar su huella de carbono comprando los créditos de CO₂, los cuáles podrá vender después en caso de ser excedentario. De este modo, las emisiones, del aprovechamiento y del transporte, no computarán en el balance de carbono de las industrias productivas, dado que éstas serán asumidas por el rematante. En el caso de la biomasa extraída, no computará para rematantes dado que, en su proceso productivo, los rematantes no consumen la madera, limitándose a hacer de nexo entre el monte y la industria.

BIOM AIS CO2

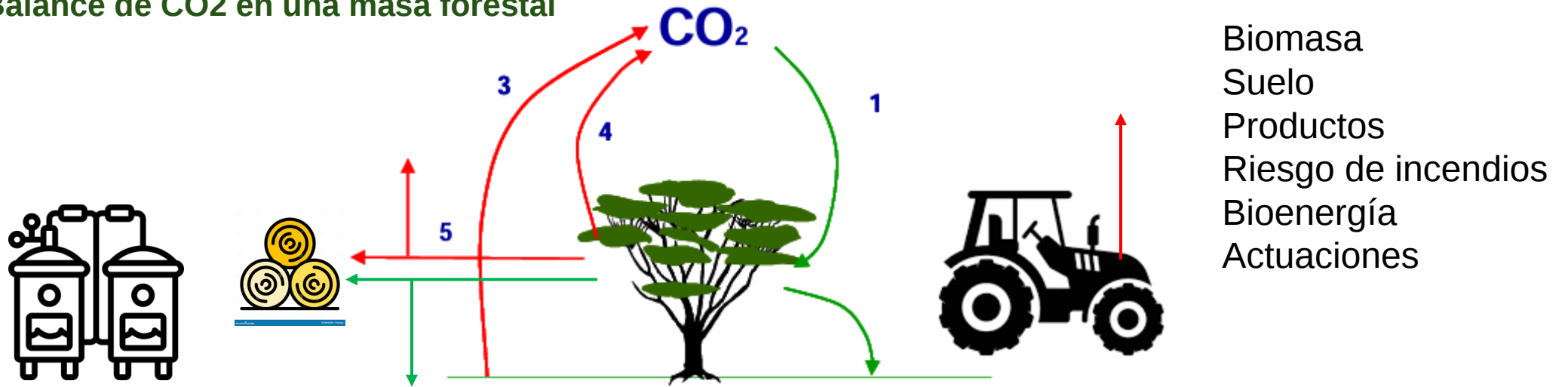
BIOM AIS CO2 es una herramienta al servicio de los técnicos forestales para contabilizar CO2, integrada en **@Bikenta**. **Aplica una metodología estandarizada para la cuantificación y seguimiento de la absorción de CO2 y reducción de emisiones**, acorde con el reglamento europeo y con la normativa ISO, que utiliza metodologías de cuantificación, previamente aprobadas por la comunidad científica, y que se han estandarizado para su uso en Galicia poniendo en valor no sólo **el crecimiento de la masa**, sino **la obtención de productos tecnológicos de larga duración con destino estructural** y los **beneficios ambientales de la gestión**.

Es una herramienta viva que deberá ir adaptándose a las metodologías aprobadas pero que cuenta con los criterios de calidad descritos en el reglamento europeo en cuanto a **línea base, adicionalidad, emisión de las actuaciones y sostenibilidad**

¿Por qué BIOMAS CO₂?

- Existe un **contexto de cambio climático y un compromiso hacia la neutralidad** carbónica promoviendo la realización de actuaciones que incrementen la absorción de CO₂ y reduzcan las emisiones, creándose el **reglamento Europeo de Certificación de Absorciones**
- La **gestión forestal sostenible y la absorción de CO₂ en los productos de larga duración** como madera estructural, se reconocen como actuaciones para aumentar la absorción de CO₂ y generar créditos de CO₂
- La venta de esos créditos de CO₂ en **el mercado voluntario** está adquiriendo cada vez más importancia y es una oportunidad para el propietario, generando **riqueza en el rural**.
- Actualmente **existen múltiples algoritmos y modelos predictivos**, avalados por la comunidad científica, y adaptados a la zona de estudio para **cuantificar el CO₂** de diferentes componentes dentro de la “masa forestal”.
- También algoritmos de **predicción de la calidad tecnológica y rendimiento de la madera estructural** desarrollados por MaderaPlus que pueden combinarse con los modelos de evolución de Biomasa para diferentes escenarios de gestión.
- **Hay una herramienta digital en uso, a disposición de los técnicos forestales** para la gestión forestal sostenible donde se puede integrar la metodología propuesta asumiendo los estándares de la normativa ISO y con los criterios de calidad definidos en el reglamento europeo.

Balance de CO₂ en una masa forestal



- Es posible aumentar la capacidad de acumulación de **biomasa** de los sistemas forestales ya establecidos mediante la aplicación de distintas propuestas de gestión (Gracia et al. 2005; Bravo et al. 2007, Carroll et al. 2012).
- Cualquier medida de gestión orientada a obtener y utilizar **productos madereros de mayor calidad** es responsable de un aumento de la absorción y fijación de CO₂ (Lizarralde et al. 2008)
- **El carbono del horizonte orgánico del suelo** es uno de los 5 grandes reservorios de carbono en los bosques
- **La probabilidad de que se inicie un incendio** forestal en una masa gestionada es muy baja frente a una masa sin gestionar

Descripción de la herramienta BIOMÁISCO₂

- La herramienta BiomaisCo₂ usa un enfoque parecido a CO₂FIX y **cuantifica las reservas y los flujos de carbono** en la biomasa forestal, la materia orgánica del suelo y la cadena de productos de la madera. Se compone de los siguientes módulos: **Biomasa, Suelo, Productos, Bioenergía, Incendios, Trabajos selvícolas.**
- Para poner en funcionamiento la aplicación es necesario completar la **información relativa a la masa** (edad actual, datos de inventario, ubicación) y **a la silvicultura propuesta en el proyecto** (año de actuación, tipo de actuación, densidad de plantación tras corta final) así como **el destino principal de la madera aprovechada** (madera estructural, sierra, tablero, papel o leñas). A partir de estos valores, **indicando la duración del proyecto, la herramienta simulará la evolución del secuestro de CO₂ en el proyecto y en la línea base.**

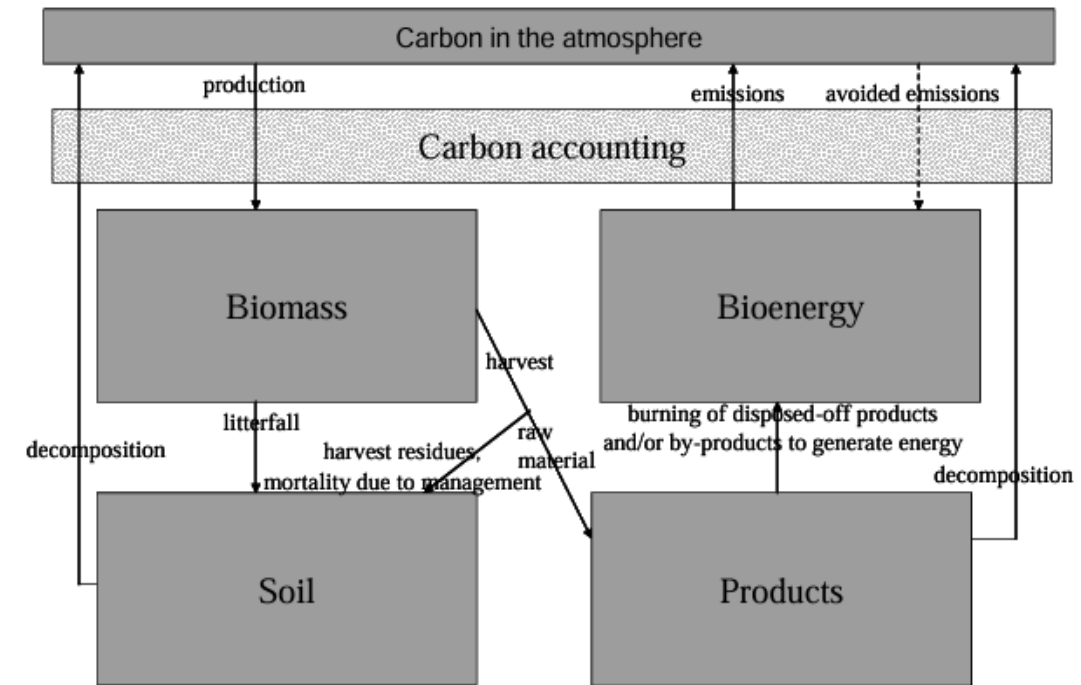


Figure 2.. The modules of CO₂FIX V 3.1.

Módulo Biomasa

- El módulo de biomasa estima la evolución de la masa empleando los modelos dinámicos de crecimientos elaborados por la UXFS de la USC para las siguientes especies: *Pinus pinaster*, *Pinus radiata*, *Pinus sylvestris*, *Eucalyptus globulus* y *Quercus robur*
- Este módulo permite **diferenciar la biomasa año a año** en cada fracción del árbol (**hojas, ramas, tronco y raíces**), lo cual se empleará para los **cálculos de biomasa en el suelo y en los productos**.
- Este módulo permitirá establecer el **volumen de biomasa forestal** en pie (tanto de la línea base como del esquema de máxima absorción propuesto) **para cada fracción de la biomasa** (hojas, ramas, raíces y fustes completos), **así como la mortalidad anual** de ésta (turnover), la cual será **utilizada para calcular los valores de carbono en el suelo**. En estas estimaciones anuales se tendrá en cuenta el **esquema silvícola, de la línea base y de la simulación de máxima captura de CO₂**.

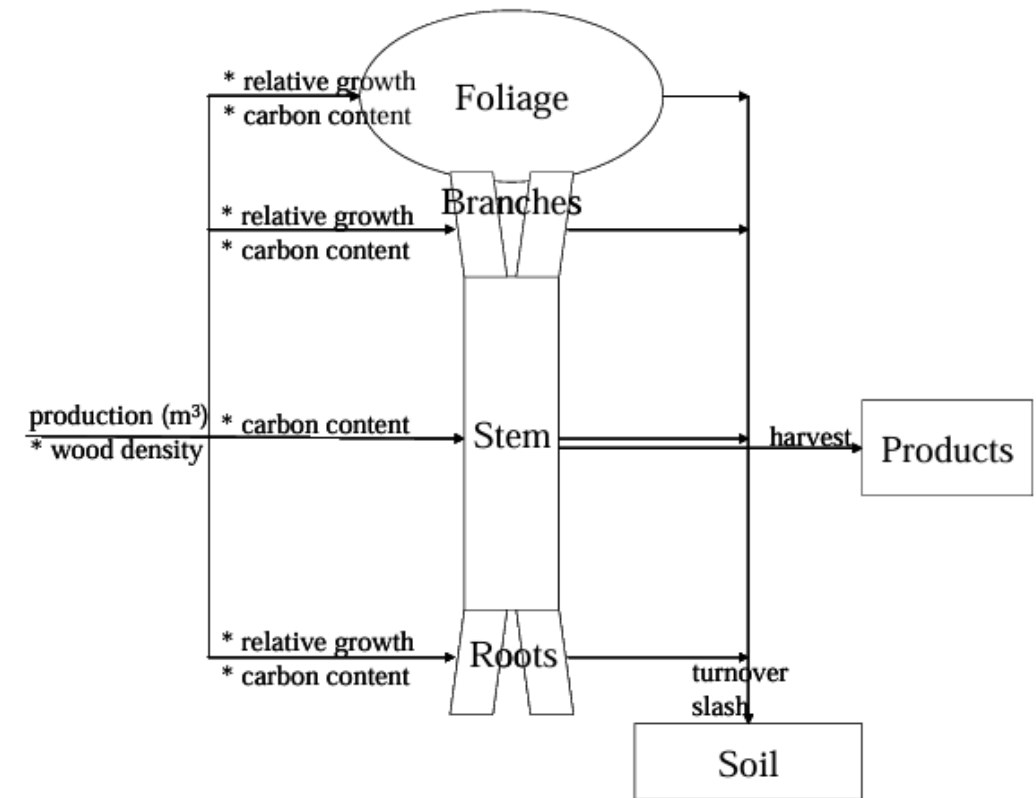
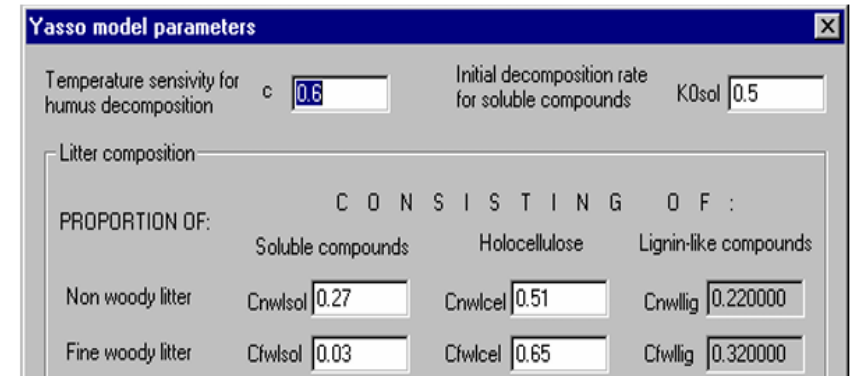


Figure 3. Schematic representation of processes and flows in the biomass module for one cohort. CO2FIX V 3.1

Módulo Suelo

- El módulo de suelo se basa, en el **modelo** dinámico de carbono del suelo **Yasso** (Liski et al.). El modelo **describe la descomposición y las dinámicas del carbono** en suelos bien drenados. La versión actual está calibrada para describir el stock total de carbono del suelo sin distinguir entre capas del suelo. Para el módulo de carbono del suelo, la hojarasca se agrupa en hojarasca no leñosa, constituida por el follaje y las raíces finas (nwl), hojarasca leñosa fina, constituida por ramas y raíces gruesas (fwl) y hojarasca leñosa gruesa (troncos y tocones).
- El modelo de suelo **considera las tasas de invasión de la materia orgánica muerta por los microorganismos y las tasas de descomposición de la materia orgánica**, considerando la **suma efectiva de las temperaturas, la evapotranspiración potencial y la precipitación acumulada en el período vegetativo**) cuyos mapas han sido desarrollados para toda Galicia en el marco de este proyecto.



Yasso model parameters

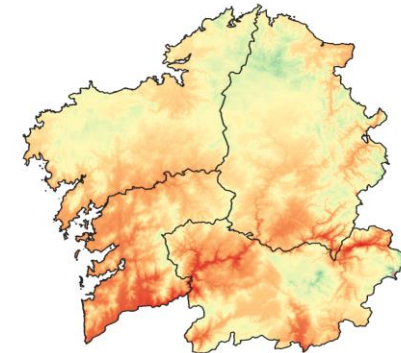
Temperature sensitivity for humus decomposition c Initial decomposition rate for soluble compounds $K0sol$

Litter composition

PROPORTION OF:	C O N S I S T I N G O F :		
	Soluble compounds	Holocellulose	Lignin-like compounds
Non woody litter	Cnwlsol 0.27	Cnwlccl 0.51	Cnwllig 0.220000
Fine woody litter	Cfwlsol 0.03	Cfwlccl 0.65	Cfwllig 0.320000

Ej pantalla del programa CO2fix en el que se base BiomáisCO2

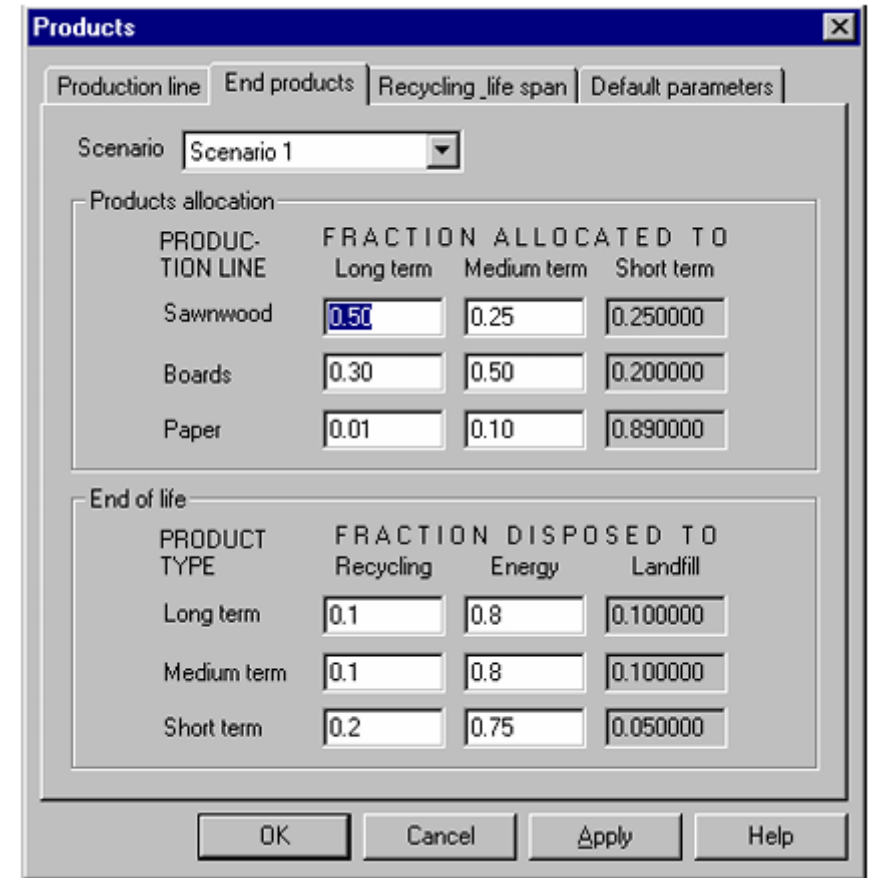
Las concentraciones de cada compartimento (solubles, celulosas y ligninas) para cada fracción de la materia orgánica de cada especie se diferencian entre coníferas y frondosas siguiendo los parámetros por defecto establecidos en el modelo CO2FIX (Schelhaas et al., 2004)).



Módulo Productos

El módulo de productos **estima el CO₂ almacenado en los productos madereros** elaborados tras las talas llevadas a cabo . El volumen de madera del fuste de los árboles extraídos, se obtiene por el módulo de biomasa en ese año concreto. Además, empleando las **ecuaciones de perfil** elaboradas por la UXFS para cada especie, es posible conocer el volumen de madera desagregado por diámetro en punta delgada saber que parte va a productos de corta, media o larga duración, lo cual determinará con que velocidad se consumen.

Así, será posible **obtener la estimación de las existencias de C y del cambio anual** de las existencias de C en el módulo de productos, **mediante la ecuación 2.8.5 de la Guía Suplementaria KP 2013**, a la que hace referencia el MITECO



PRODUCTION LINE	Long term	Medium term	Short term
Sawnwood	0.50	0.25	0.250000
Boards	0.30	0.50	0.200000
Paper	0.01	0.10	0.890000

PRODUCT TYPE	Recycling	Energy	Landfill
Long term	0.1	0.8	0.100000
Medium term	0.1	0.8	0.100000
Short term	0.2	0.75	0.050000

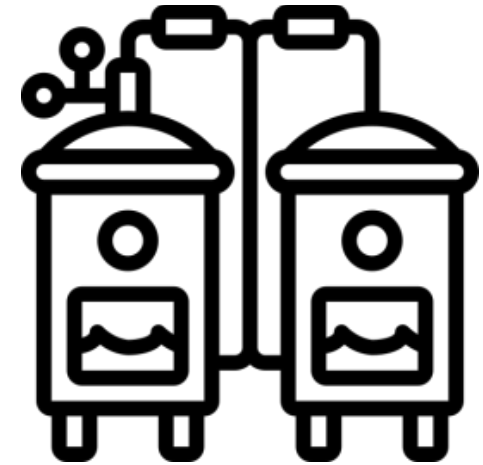
Ej pantalla del programa CO2fix en el que se base BiomáisCO2

Bioenergía

Sustituir los combustibles fósiles por bioenergía producida de manera sostenible es una forma de mitigar las emisiones de gases de efecto invernadero a la atmósfera. Tanto en la herramienta BiomaisCO₂ como en CO₂FIX, se consideran dos tipos de combustibles de biomasa: uno que proviene de residuos industriales (como productos desechados o pérdidas durante el procesamiento) y otro de los restos que se extraen del sitio forestal.

La sustitución de combustibles fósiles por biomasa conduce a una mitigación permanente de los gases de efecto invernadero (GEI). Por lo tanto, podemos considerar los GEI mitigados acumulativamente como un aumento en las reservas de carbono en los bosques.

Para nuestra herramienta, el combustible a sustituir será siempre el gasóleo C. El MITECO propone unos factores de emisión para la biomasa viva y para el gasóleo C, que son los empleados en la herramienta.



Modulo incendios

Se estima el riesgo de incendios en base a la fórmula del IPCC multiplicando por una probabilidad de incendio (Cervera et al., 2022). Dicha metodología se basa en la siguiente fórmula de cálculo:

$$\text{tCO}_2 \text{ IF} = (\text{Ton MS de Biomasa aérea} \times \text{Fcomb} \times \text{FE}) \times \text{Probabilidad Incendio}$$

La parte entre paréntesis de la ecuación se toma del IPCC 2006 para estimar las emisiones de GHG por incendios forestales (multiplicada por la superficie que no se indica aquí porque los cálculos se refieren a la ha).

La probabilidad de incendio **se calcula sabiendo las ha quemadas medias anualmente en el Distrito** (período 2016-2022 según estadísticas oficiales) y las ha de superficie forestal en el Distrito.

La siguiente tabla indica los factores de combustión en función de la intensidad esperada (De Santis et al., 2010) y los factores de emisión de CO₂ (Miranda et al., 2005), según tipo de vegetación.

Tipo de vegetación	Factor de combustión (FComb) (% biomasa consumida)				Factor emisión CO ₂ (FE) (tCO ₂ tms ⁻¹)
	Clase de intensidad del fuego (<i>fire severity</i>)				
	Baja (CFL < 1,2 m)	Media (CFL = 1,2-2,4 m)	Alta (CFL = 2,4-3,4 m)	Muy alta (CFL => 3,4 m)	
Matorral	0,71	0,84	0,89	0,95	1,477
Coníferas	0,25	0,47	0,56	0,65	1,627
Frondosas	0,25	0,40	0,48	0,56	1,393

Modulo actuaciones

La maquinaria utilizada en las distintas actuaciones selvícolas causa de emisiones de CO₂

Para cuantificar estas emisiones se ha realizado la captura de datos durante un año completo sobre más de 100 actuaciones, analizando el consumo de litros de gasoil por hora y posteriormente referenciándolo a litros por ha o por Tonelada

Se diferenciaron:

- Actuaciones selvícolas (trituración de restos, rareos, preparación terreno, desbroce: podas.....)
- Aprovechamiento forestal (corta, saca, carga,.....)
- Transporte

En este caso las emisiones de CO₂ sólo serán contabilizadas en la fase de ejecución, pero serán despreciadas en la fase de planificación

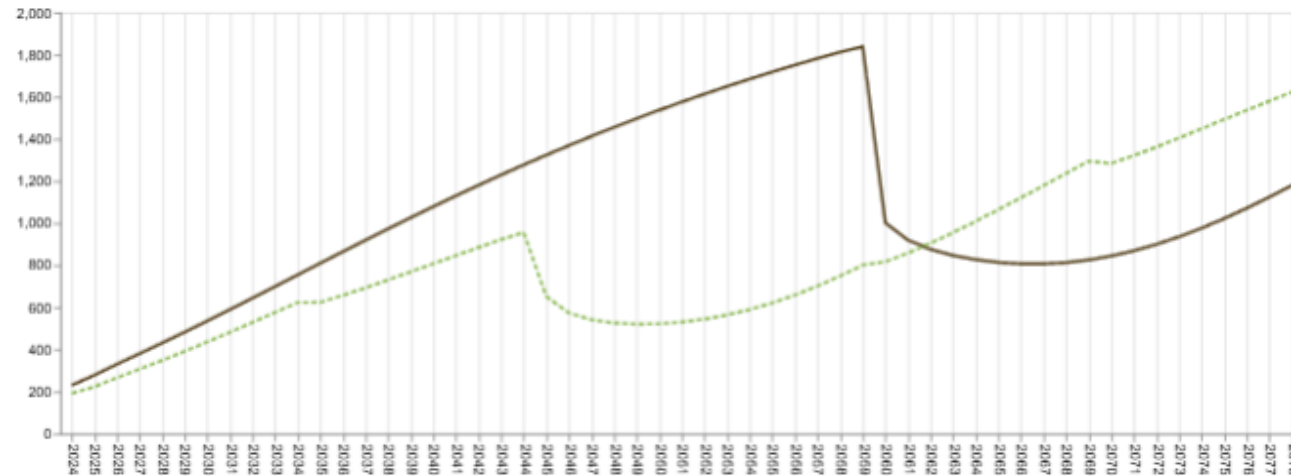
Para contabilizar estas emisiones el usuario deberá asociar una dificultad de mecanización al rodal y elegir la especie y dentro de las actuaciones si el modo de realización es manual, mecanizado o mixto

Tras la corta de la madera, se podrá emitir un certificado con el CO₂ asociado a cada T de madera (incluyendo la parte proporcional de biomasa extraída y la aportada al suelo)



Ejemplo de balance de absorción

C. PLAN DE ABSORCIÓN



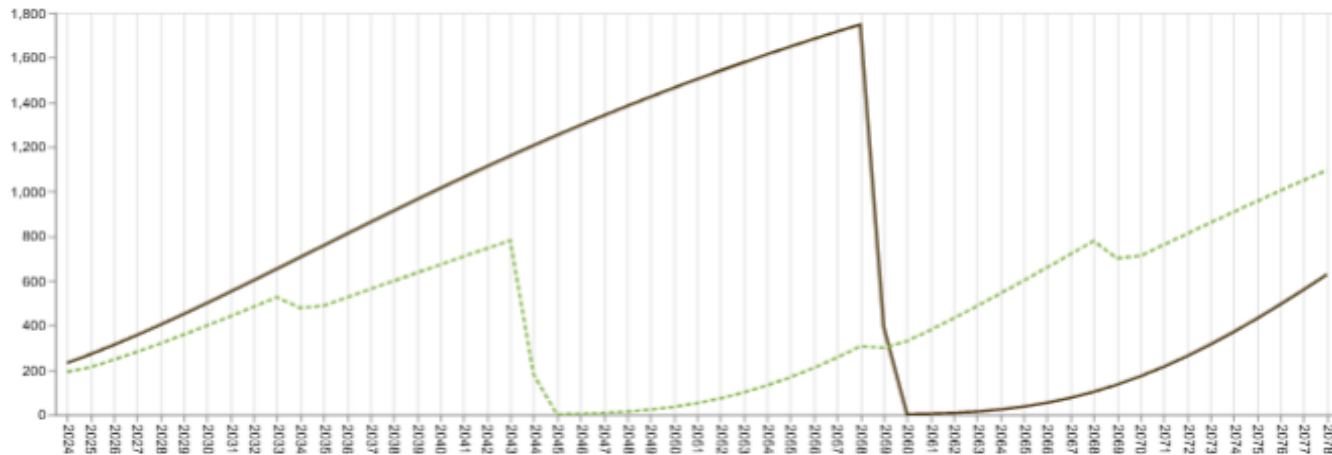
Total

1179 T CO₂
LÍNEA BASE

1622 T CO₂
PROYECTO SELVÍCOLA

443 T CO₂
ADICIONALIDAD

C. PLAN DE ABSORCIÓN



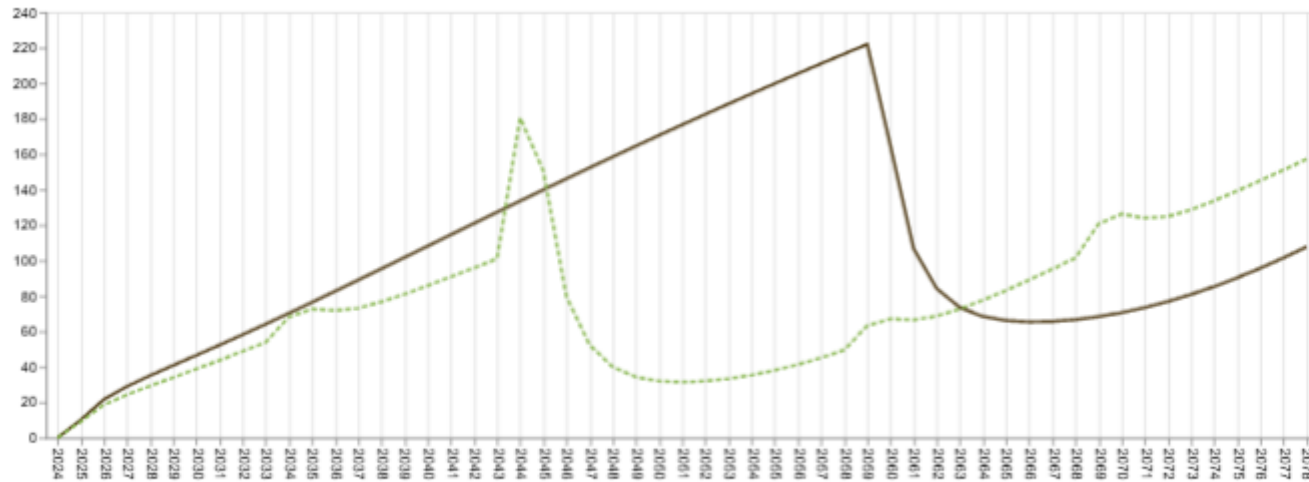
Biomasa

628 T CO₂
LÍNEA BASE

1095 T CO₂
PROYECTO SELVÍCOLA

467 T CO₂
ADICIONALIDAD

C. PLAN DE ABSORCIÓN

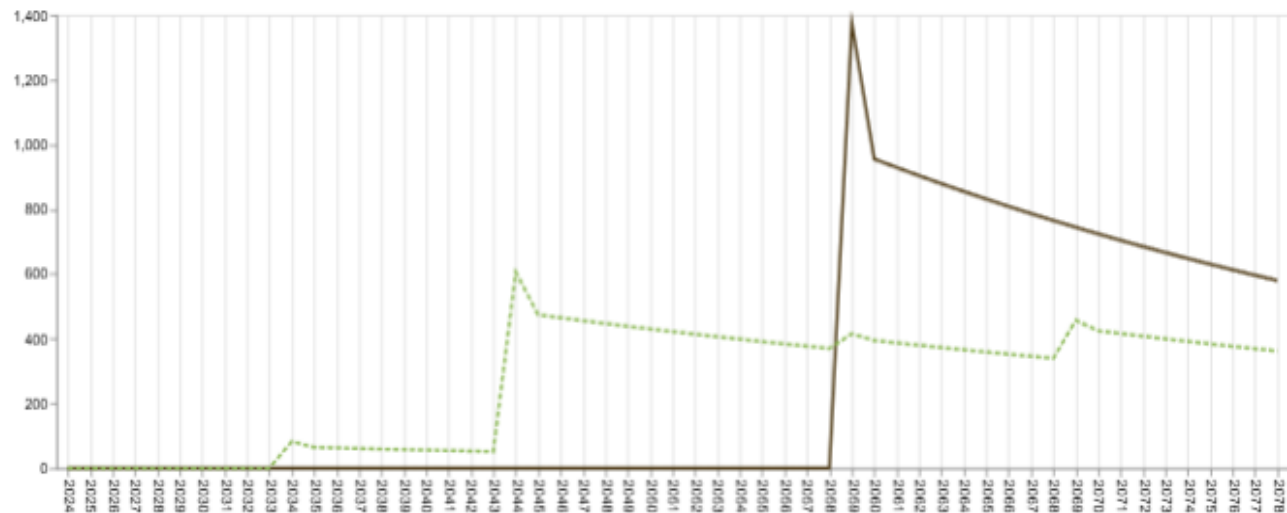


108 T CO₂
LÍNEA BASE

157 T CO₂
PROYECTO SELVÍCOLA

49 T CO₂
ADICIONALIDAD

C. PLAN DE ABSORCIÓN



580 T CO₂
LÍNEA BASE

362 T CO₂
PROYECTO SELVÍCOLA

-218 T CO₂
ADICIONALIDAD

Conclusiones

- **El CO2 es un recurso más del monte** que **deberá ser gestionado por un profesional forestal** y es una oportunidad para promover una selvicultura orientada a producto, **aportando riqueza al propietario y generando empleo local**
- La **herramienta** está pensada **para dar apoyo y servicio a los técnicos forestales** (ingenieros forestales y de montes) **responsables de la gestión forestal** de la masa sobre la que se pretende actuar
- Biomais CO2 es una **herramienta viva** que deberá ir **adaptándose a las metodologías aprobadas**, pero cuenta con los criterios de calidad descritos en el reglamento europeo en cuanto a **línea base, adicionalidad, emisión de las actuaciones y sostenibilidad**
- Los cálculos se realizan **a partir de los datos de inventario, recogidos por el técnico** con mediciones pie a pie o a partir de datos LiDAR y algoritmos. La correcta trazabilidad y veracidad de los datos es responsabilidad del técnico y deberá ser verificada y auditada
- La herramienta está **preparada para realizar el seguimiento de la ejecución del proyecto** y captura de **datos de las actuaciones** como de **los datos de inventario**.
- En el caso de una corta, será posible calcular el **CO2 asociado a la madera en rollo**

Proyecto financiado por



Fondo Europeo Agrícola
de Desenvolvemento Rural:
Europa inviste no rural



GOBIERNO
DE ESPAÑA

MINISTERIO
DE AGRICULTURA, PESCA
Y ALIMENTACIÓN

Gracias por su atención

Esther Merlo Sánchez
MADERA PLUS
maderaplus@maderaplus.es



BIOMÁIS CO2 se desarrolló en el marco de un grupo operativo da Asociación Europea de Innovación (AEI), cofinanciado con el Fondo Europeo Agrícola de Desarrollo Rural (FEADER) en el marco del Programa de Desarrollo Rural (PDR) de Galicia 2014-2020.

Presupuesto: 135.396,96 €.