

Documentació API Grup A

Ferran Casanovas i Arnau Parcerisa

18 de gener de 2024

Índex

1	Introducció	1
2	Formats de les peticions	2
3	Peticions POST	2
3.1	/assignaDispositiuUsuari	2
3.2	/desassignaDispositiu	3
3.3	/inserirDada	3
3.4	/inserirDispositiu	4
3.5	/inserirEstatReg	5
3.6	/inserirUsuari	5
3.7	/modificaContrasenya	6
3.8	/modificaLlindars	7
3.9	/verificaLogIn	7
3.10	/verificaUsuari	8
4	Peticions GET	10
4.1	/obtenirCanvisReg	10
4.2	/obtenirDadesDispositius	11
4.3	/obtenirDispositius	13
4.4	/obtenirDispositiusDesassignats	14
4.5	/comprovacioDispositiuInserit	14
4.6	/obtenirUsuaris	15
4.7	/obtenirUltimReg	16
4.8	/obtenirUltimRegUsuari	17
4.9	/obtenirUltimaDadaDispositiu	18
4.10	/obtenirModificacionsLlindars	19

1 Introducció

En aquest document s'hi pot trobar tota la documentació de la API creada pel grup A, format per l'Arnau Parcerisa, el Xavier Massana i el Ferran Casanovas.

Aquesta API permet, mitjançant mètodes GET i POST, consultar i inserir dades a la base de dades *MySQL* dissenyada. Per a la programació d'aquesta API s'ha utilitzat Python3 i Flask.

2 Formats de les peticions

En cadascuna de les peticions, s'especifica com ha de ser el format i quines dades contindrà la resposta. En cas de que la petició sigui correcta i no hi hagin errors durant l'execució, el codi de resposta serà 200 i el seu format serà diferent en cada petició, donat que en cadascuna es voldran obtenir dades diferents. No obstant, sempre s'indicarà que no hi han hagut errors retornant una petició amb el format següent:

```
{
  {
    "success" = True,
    ...
  }
}
```

En cas que la petició sigui incorrecta, ja sigui perquè aquesta és sintàcticament incorrecta, perquè no s'especifiquen les dades necessàries o perquè el client no està autoritzat per accedir i/o modificar el recurs, el codi de resposta serà 400. Si es produeix algun error durant l'execució el codi de resposta serà 500. En ambdós casos, el format de la resposta serà el següent:

```
{
  "success" = False,
  "error" = "descripció del error"
}
```

3 Peticions POST

A continuació es descriuen les funcionalitats de totes les peticions de la API que utilitzen el mètode POST, incloent una breu descripció, característiques de seguretat (si n'hi han), el format de la petició i el format de la resposta per a cada petició. Cal destacar que en tots els casos s'espera que en el cos de la petició s'hi inclogui un objecte `json`.

3.1 /assignaDispositiuUsuari

Endpoint de Flask per assignar un dispositiu a un usuari. Només permet assignar-lo si el dispositiu està desassignat, és a dir, que no està assignat a cap altra usuari.

3.1.1 Format de la petició

```
{
  "idDispositiu": (int),
  "idUsuari": (int),
  "nomDispositiu": (str)
}
```

En aquest cas, es poden ocultar camps del json. Els camps del json `idDispositiu`, `idUsuari` i `nomDispositiu` son **OBLIGATORIS**.

3.1.2 Format de la resposta

```
{
  "success" = True
  "dades":
  {
    "idHardcode": (str)
  }
}
```

3.2 /desassignaDispositiu

Endpoint de Flask per desassignar un dispositiu a un usuari. Només es pot desassignar un dispositiu si aquest està assignat a un usuari. A més, s'esborren totes les dades del dispositiu de les taules de dades corresponents a les dades de humitat i temperatura (`dadesDispositius`) i a la de canvis d'estat de reg (`canvisReg`).

3.2.1 Format de la petició

```
{
  "idDispositiu": (int),
  "idUsuari": (int)
}
```

Els camps de la petició `idDispositiu` i `idUsuari` són obligatoris.

3.2.2 Format de la resposta

```
{
  "success" = True
}
```

3.3 /inserirDada

Endpoint de Flask per inserir dades a la taula `dadesDispositius`.

3.3.1 Format de la petició

```
{
  "idHardcode": (str),
  "dadaHum": (decimal),
  "dadaTemp": (decimal),
}
```

En aquest cas, es poden ocultar camps del del **json**. No es pot inserir dades sense que no estigui assignat a cap usuari. **Tots els camps del json són OBLIGATORIS.**

3.3.2 Format de la resposta

```
{
  "success" = True
}
```

3.4 /inserirDispositiu

Endpoint de Flask per inserir dades a la taula dispositius. Permet inserir un dispositiu no assignat a cap usuari. El fet d'inserir un dispositiu no assignat es fa per tal de facilitar la feina del **front-end** per assignar un nou dispositiu a un usuari. Tot i així, si que és obligatori afegir-li un id que el permeti identificar amb el hardware, anomenat **idHardcode**.

3.4.1 Format de la petició

```
{
  "idHardcode": (str),
  "idUsuariPropietari": (int) / None,
  "nomDispositiu": (str) / None
  "llindarMinimReg": (decimal) / None,
  "llindarMaximReg": (decimal) / None
}
```

En aquest cas, es poden ocultar camps del del **json**. Si es vol inserir un dispositiu que no estigui assignat a cap usuari, s'ha enviar un **json** que només contingui el id del hardware. **Tots els camps del json són OPCIONALS en excepció del idHardcode que és OBLIGATORI.**

3.4.2 Format de la resposta

Quan el dispositiu s'insereix correctament, s'especifica el id que se li ha assignat al dispositiu inserit.

```
{
  "success": True,
  "idDispositiu": (int),
  "idHardcode": (str)
}
```

3.5 /inserirEstatReg

Endpoint de Flask per inserir dades a la taula canvisReg. Aquest només insereix els canvis, és a dir només insereix la dada si l'últim estat del reg del dispositiu és diferent.

3.5.1 Format de la petició

```
{
  "idHardcode": (int),
  "estatReg": (bool)
}
```

Si el cos de la petició no conté un json amb els camps idHardcode i estatReg, les dades no s'inseriran. **Tots els camps del json són OBLIGATORIS.**

3.5.2 Format de la resposta

Si tot funciona correctament, s'especifica si la dada s'ha inserit o no depenent de si aquesta és diferent o no a l'últim estat del reg del dispositiu guardat.

```
{
  "success" = True,
  "dadaInserida": (bool)
}
```

3.6 /inserirUsuari

Endpoint de Flask per inserir dades a la taula usuaris. Al inserir un usuari, s'insereix a la base de dades la data utf de creació de l'usuari, aquesta data no s'ha d'especificar en la petició.

3.6.1 Seguretat

S'ha d'enviar la contrasenya sense aplicar cap funció de hash, ja que la mateixa petició és la que s'encarrega d'aplicar-lo.

3.6.2 Format de la petició

```
{
  "email": (str),
  "nomUsuari": (str),
  "contrasenya": (str)
}
```

Si el cos de la petició no conté un json amb els camps `email`, `nomUsuari` i `contrasenya`, les dades de l'usuari no s'inseriran. **Tots els camps del json són OBLIGATORIS.**

3.6.3 Format de la resposta

Quan l'usuari s'insereix correctament, s'especifica el id que se li ha assignat a l'usuari insertat.

```
{
  "success": True,
  "idUsuariInsertat": (int)
}
```

3.7 /modificaContrasenya

Endpoint de Flask per modificar la contrasenya d'un usuari. Per tal de modificar-la és necessari indicar la contrasenya actual i que aquesta sigui correcta.

3.7.1 Seguretat

S'ha d'enviar tant la contrasenya actual com la nova contrasenya sense aplicar cap funció de hash, donat que la mateixa petició s'encarrega de:

- Aplicar la funció de hash a la contrasenya actual indicada i comprovar que coincideixi amb la guardada a la base de dades.
- Guardar la nova contrasenya amb la funció de hash aplicada.

3.7.2 Format de la petició

```
{
  "idUsuari": (int),
  "contrasenya": (str),
  "novaContrasenya": (str)
}
```

Si el cos de la petició no conté un json amb els camps `idUsuari`, `contrasenya` i `novaContrasenya`, la contrasenya no es modificarà. **Tots els camps del json són OBLIGATORIS.**

3.7.3 Format de la resposta

Indica si s'ha modificat la contrasenya correctament.

```
{
  "success" = True
}
```

3.8 /modificaLlindars

Endpoint de Flask per modificar els llindars de reg d'un dispositiu. S'emmagatzemen els llindars modificats per si es demana en la petició GET /obtenirModificacionsLlindars. Si ja hi ha algun canvi de llindar entrat, se'l substitueix.

3.8.1 Format de la petició

```
{
  "idDispositiu": (int),
  "llindarMinimReg": (decimal),
  "llindarMaximReg": (decimal)
}
```

Si el cos de la petició no conté un json amb els camps `idDispositiu`, `llindarMinimReg` i `llindarMaximReg`, la petició donarà a un error amb codi 400. **Els camps del json `idDispositiu`, `llindarMinimReg` i `llindarMaximReg` són OBLIGATORIS.**

3.8.2 Format de la resposta

Indica si s'han modificat els llindars correctament.

```
{
  "success" = True
}
```

3.9 /verificaLogIn

Endpoint de Flask per obtenir si les credencials de log in es corresponen a les credencials de la base de dades. Si troba les credencials, retorna el identificador de l'usuari per facilitar el logIn.

3.9.1 Seguretat

S'ha d'enviar la contrasenya sense aplicar cap funció de hash, ja que la mateixa petició s'encarrega d'aplicar-la i comprovar que coincideixi amb la guardada a la base de dades.

3.9.2 Format de la petició

```
{
  "emailUsuari": (str),
  "contrasenya": (str)
}
```

Si el cos de la petició no conté un json amb els camps emailUsuari i contrasenya, la petició donarà a un error amb codi 400. **Tots els camps del json són OBLIGATORIS.**

3.9.3 Format de la resposta

Indica si s'ha trobat una coincidència de les credencials a la base de dades o no i retorna l'identificador de l'usuari per facilitar el LogIn. També retorna si l'usuari està verificat o no.

```
{
  "success" = True,
  "credencialsTrobades": (bool),
  "idUsuari": (int) / None,
  "usuariVerificat": (bool)
}
```

3.10 /verificaUsuari

Endpoint de Flask per verificar el codi de verificació d'un usuari.

3.10.1 Format de la petició

```
{
  "idUsuari": (int),
  "codi": (str)
}
```

Si el cos de la petició no conté un json amb els camps idUsuari i codi, la petició donarà a un error amb codi 400. **Tots els camps del json són OBLIGATORIS.**

3.10.2 Format de la resposta

Indica si s'ha trobat una coincidència del codi de verificació a la base de dades o no i un boolea que indica si el codi introduït i el de la base de dades eren els mateixos i, per tant, l'usuari s'ha verificat (**True**) o, en cas contrari, els codis no coincideixen i, per tant, l'usuari no s'ha verificat (**False**).


```
{  
  "success" = True,  
  "usuariVerificat": (bool)  
}
```

4 Peticions GET

A continuació es descriuen les funcionalitats de totes les peticions de la API que utilitzen el mètode GET, incloent una breu descripció, descripció dels paràmetres de la petició, el format de la resposta i exemples per a cada petició. Cal destacar que el format de resposta de totes les peticions GET serà el següent:

```
{
    "success" = True,
    "dades":
    [
        ...
    ]
}
```

On dintre de la llista **dades** s'hi emmagatzemaran les dades que es volen consultar, generalment dintre de diccionaris on cada diccionari es correspon a una fila de la base de dades. Les dades que es retornaran en cada petició i el seu format s'especificarà en la descripció de cadascuna d'elles.

4.1 /obtenirCanvisReg

Endpoint de Flask per obtenir dades de la taula `canvisReg`, on estan emmagatzemats els canvis en l'estat del reg.

4.1.1 Paràmetres de la petició

- **idUsuari (opcional)**: identificador del usuari del qual es vol obtenir els canvis en l'estat del reg dels seus dispositius assignats.
- **idDispositiu (opcional)**: identificador del dispositiu del qual es volen obtenir els canvis en l'estat del reg.
- **dataInici (opcional)**: data a partir de la qual es volen obtenir els canvis en l'estat del reg.
- **dataFi (opcional)**: data fins a la qual es volen obtenir els canvis en l'estat del reg.

Si no s'especifiquen cap dels paràmetres, s'obtindran totes les dades emmagatzemades en la taula `canvisReg`. Si s'especifiquen els paràmetres `idUsuari` i `idDispositiu`, es filtrarà per l'identificador del dispositiu. Els paràmetres `dataInici` i/o `dataFi` han d'anar acompanyats del paràmetre `idUsuari` o `idDispositiu`, en cas contrari s'obtindran totes les dades sense filtrar per data. Es pot especificar el paràmetre `dataInici` sense especificar `dataFi`, però si s'especifica `dataFi` sense especificar `dataInici` aquest s'ignorarà.

4.1.2 Format de la resposta

```
{
  "success" = True,
  "dades":
  [
    {
      "idDispositiu": (int),
      "dataHora": (datetime),
      "estatReg": (bool)
    },
    ...
  ]
}
```

4.1.3 Exemples de peticions

- Obtenir totes les dades:

```
GET /obtenirCanvisReg
```

- Obtenir totes les dades dels dispositius assignats a un usuari en concret:

```
GET /obtenirCanvisReg?idUsuari=1
```

- Obtenir totes les dades d'un dispositiu:

```
GET /obtenirCanvisReg?idDispositiu=1
```

- Obtenir totes les dades dels dispositius assignats a un usuari indicant data inici i data fi:

```
GET /obtenirCanvisReg?idUsuari=1&dataInici=2023-11-16 00:01:00 &dataFi=31-12-2023 23:59:00
```

- Obtenir totes les dades d'un dispositiu indicant data inici i data fi:

```
GET /obtenirCanvisReg?idDispositiu=1&dataInici=01-01-2023 00:01:00 &dataFi=31-12-2023 23:59:00
```

4.2 /obtenirDadesDispositius

Endpoint de Flask per obtenir dades de la taula `dadesDispositius`, on estan emmagatzemats els valors de temperatura i d'humitat.

4.2.1 Paràmetres de la petició

- **idUsuari (opcional)**: identificador del usuari del qual es volen obtenir les dades dels seus dispositius assignats.

- **idDispositiu (opcional)**: identificador del dispositiu del qual es volen obtenir les dades.
- **dataInici (opcional)**: data a partir de la qual es volen obtenir les dades.
- **dataFi (opcional)**: data fins a la qual es volen obtenir les dades.

Si no s'especifiquen cap dels paràmetres, s'obtindran totes les dades emmagatzemades en la taula `dadesDispositius`. Si s'especifiquen els paràmetres `idUsuari` i `idDispositiu`, es filtrarà per l'identificador del dispositiu. Els paràmetres `dataInici` i/o `dataFi` han d'anar acompanyats del paràmetre `idUsuari` o `idDispositiu`, en cas contrari s'obtindran totes les dades sense filtrar per data. Es pot especificar el paràmetre `dataInici` sense especificar `dataFi`, però si s'especifica `dataFi` sense especificar `dataInici` aquest s'ignorarà.

4.2.2 Format de la resposta

```
{
  "success" = True,
  "dades":
  [
    {
      "idDispositiu": (int),
      "dataHora": (datetime),
      "dadaHum": (decimal),
      "dataTemp": (decimal)
    },
    ...
  ]
}
```

4.2.3 Exemples de peticions

- Obtenir totes les dades:
`GET /obtenirDadesDispositius`
- Obtenir totes les dades dels dispositius assignats a un usuari en concret:
`GET /obtenirDadesDispositius?idUsuari=1`
- Obtenir totes les dades d'un dispositiu:
`GET /obtenirDadesDispositius?idDispositiu=1`
- Obtenir totes les dades dels dispositius assignats a un usuari indicant data inici i data fi:
`GET /obtenirDadesDispositius?idUsuari=1&dataInici=2023-11-16 00:01:00
&dataFi=31-12-2023 23:59:00`

- Obtenir totes les dades d'un dispositiu indicant data inici i data fi:

```
GET /obtenirDadesDispositius?idDispositiu=1&dataInici=01-01-2023 00:01:00
&dataFi=31-12-2023 23:59:00
```

4.3 /obtenirDispositius

Endpoint de Flask per obtenir dades de la taula dispositius.

4.3.1 Paràmetres de la petició

- **idUsuari (opcional)**: id de l'usuari del qual es volen obtenir les dades dels seus dispositius assignats.
- **nomDispositiu (opcional)**: nom del dispositiu del qual es volen obtenir les dades. Sempre ha d'anar acompanyat d'un idUsuari, ja que diferents usuaris poden tenir dispositius amb el mateix nom.
- **idDispositiu (opcional)**: id del dispositiu del qual es poden obtenir les dades.

Si no s'especifica cap dels paràmetres, s'obtidran les dades de tots els dispositius. Si s'especifica el paràmetre `idDispositiu`, es filtrarà per aquest encara que s'especifiquin més paràmetres.

4.3.2 Format de la resposta

```
{
  "success" = True,
  "dades":
  [
    {
      "id": (int),
      "idHardcode": (str),
      "idUsuariPropietari": (int),
      "nomDispositiu": (str),
      "llindarMinimReg": (decimal),
      "llindarMaximReg": (decimal)
    },
    ...
  ]
}
```

4.3.3 Exemples de peticions

- Obtenir totes les dades:
`GET /obtenirDispositius`

- Obtenir les dades dels dispositius d'un usuari a partir d'un idUsuari:
`GET /obtenirDispositius?idUsuari=1`
- Obtenir les dades d'un dispositiu en concret a partir del id:
`GET /obtenirDispositius?idDispositiu=1`
- Obtenir les dades d'un dispositiu en concret a partir del id del usuari i el nom de dispositiu:
`GET /obtenirDispositius?idUsuari=1&nomDispositiu=dispo1`

4.4 /obtenirDispositiusDesassignats

Endpoint de Flask per obtenir els dispositius que no estan assignats a cap usuari, és a dir, que la columna `idUsuariPropietari` = `NULL`.

4.4.1 Paràmetres de la petició

Aquesta petició no té paràmetres, donat que al realitzar-la s'obtindran tots els dispositius desassignats.

4.4.2 Format de la resposta

```
{
  "success" = True,
  "dades":
  [
    {
      "idDispositiu": (int),
      "idHardcode": (str)
    },
    ...
  ]
}
```

4.4.3 Exemples de peticions

- Obtenir tots els dispositius desassignats:
`GET /obtenirDispositiusDesassignats`

4.5 /comprovacioDispositiuInserit

Endpoint de Flask per comprovar si un dispositiu ha sigut inserit a la base de dades i està assignat.

4.5.1 Paràmetres de la petició

- **idHardcode (obligatori)**: identificador únic del dispositiu.

Si no s'especifica el paràmetre **idHardcode**, la petició donarà lloc a un error amb codi 400. **El paràmetre de la petició és OBLIGATORI.**

4.5.2 Format de la resposta

```
{
  "success": True,
  "dispositiuTrobat": (bool),
  "assignat": (bool)
}
```

4.5.3 Exemples de peticions

- Comprovar si un dispositiu està inserit a la base de dades:

GET /comprovacioDispositiuInserit?idHardcode=A

4.6 /obtenirUsuaris

Endpoint de Flask per obtenir dades de la taula usuaris. Permet obtenir les dades d'un usuari en concret o de tots els usuaris. Cal tenir en compte que mitjançant aquest mètode mai es podran obtenir les contrasenyes dels usuaris, ni tan sols amb la funció hash aplicada.

4.6.1 Paràmetres de la petició

- **idUsuari (opcional)**: identificador del usuari del qual es volen obtenir les dades.
- **emailUsuari (opcional)**: correu electrònic del usuari del qual es volen obtenir les dades.

Si no s'especifica cap dels dos paràmetres, s'obtindran les dades de tots els usuaris. Si s'especifiquen els dos paràmetres, es filtrarà per l'identificador de l'usuari.

4.6.2 Format de la resposta

```
{
  "success" = True,
  "dades":
  [
    {
      "id": (int),
```

```

        "email": (str),
        nomUsuari: (str),
        dataCreacioUsuari: (datetime)
    },
    ...
]
}

```

4.6.3 Exemples de peticions

- Obtenir les dades de tots els usuaris:
GET /obtenirUsuaris
- Obtenir les dades d'un usuari en concret mitjançant el id:
GET /obtenirUsuaris?idUsuari=1
- Obtenir les dades d'un usuari en concret mitjançant el email:
GET /obtenirUsuaris?emailUsuari=usuari@exemple.com

4.7 /obtenirUltimReg

Endpoint de Flask per obtenir les dades del últim reg d'un dispositiu.

4.7.1 Paràmetres de la petició

- **idDispositiu (obligatori)**: identificador del dispositiu del qual es volen obtenir les dades de l'últim reg.

Si no s'especifica el paràmetre idDispositiu, la petició donarà lloc a un error amb codi 400. **El paràmetre de la petició és OBLIGATORI.**

4.7.2 Format de la resposta

```

{
    "success" = True,
    "dades":
    [
        {
            "dataHoraInici": (datetime) / "",
            "dataHoraFi": (datetime) / ""
        }
    ]
}

```

Es poden donar tres opcions, que donen tres respostes diferents:

- **S'ha començat i s'ha acabat de regar:** l'últim reg s'ha finalitzat, pel que la bomba es troba parada. `dataHoraInici` indica el moment en el qual s'ha començat a regar i `dataHoraFi` indica el moment en el qual s'ha finalitzat el reg.
- **S'ha començat a regar i no s'ha acabat el reg:** l'últim reg no s'ha finalitzat, pel que la bomba es troba engegada. `dataHoraInici` indica el moment en el qual s'ha començat a regar i `dataHoraFi` és un string buit.
- **No s'ha regat mai:** no es tenen dades de cap reg del dispositiu. Tant `dataHoraInici` com `dataHoraFi` són strings buits.

4.7.3 Exemples de peticions

- Obtenir les dades de l'últim reg d'un dispositiu:
`GET /obtenirUltimReg?idDispositiu=1`

4.8 /obtenirUltimRegUsuari

Endpoint de Flask per obtenir les dades del últim reg de tots els dispositiu d'un usuari.

4.8.1 Paràmetres de la petició

- **idUsuari (obligatori):** identificador del usuari del qual es volen obtenir les dades de l'últim reg de tots els seus dispositius.

Si no s'especifica el paràmetre `idUsuari`, la petició donarà lloc a un error amb codi 400. **El paràmetre de la petició és OBLIGATORI.**

4.8.2 Format de la resposta

```
{
  "dades":
  [
    {
      "idDispositiu": (int) / "",
      "nomDispositiu": (str) / "",
      "dataHoraInici": (datetime) / "",
      "dataHoraFi": (datetime) / ""
    },
    {
      "idDispositiu": (int) / "",
      "nomDispositiu": (str) / "",
      "dataHoraInici": (datetime) / "",
      "dataHoraFi": (datetime) / ""
    }
  ]
}
```

```

    }
  ]
  "success": True,
}

```

Es poden donar tres opcions, que donen tres respostes diferents:

- **S'ha començat i s'ha acabat de regar:** l'últim reg s'ha finalitzat, pel que la bomba associada al dispositiu es troba parada. `dataHoraInici` indica el moment en el qual s'ha començat a regar i `dataHoraFi` indica el moment en el qual s'ha finalitzat el reg. `idDispositiu` indica al dispositiu on està connectada la bomba i `nomDispositiu` indica el seu nom donat.
- **S'ha començat a regar i no s'ha acabat el reg:** l'últim reg no s'ha finalitzat, pel que la bomba es troba engegada. `dataHoraInci` indica el moment en el qual s'ha començat a regar i `dataHoraFi` és un string buit. `idDispositiu` indica al dispositiu on està connectada la bomba i `nomDispositiu` indica el seu nom donat.
- **No s'ha regat mai:** no es tenen dades de cap reg de cap dispositiu de l'usuari. Tant `dataHoraInici` com `dataHoraFi` com `idDispositiu` com `nomDispositiu` són strings buits.

4.8.3 Exemples de peticions

- Obtenir les dades de l'últim reg dels dispositius d'un usuari:

```
GET /obtenirUltimRegUsuari?idUsuari=1
```

4.9 /obtenirUltimaDadaDispositiu

Endpoint de Flask per obtenir les últimes dades de la taula `dadesDispositius`. Això inclou obtenir les últimes dades d'un dispositiu en concret o obtenir totes les dades dels dispositius assignats a un usuari.

4.9.1 Paràmetres de la petició

- **idUsuari (opcional):** identificador del usuari del qual es vol obtenir la última dada de cadascun dels dispositius que té assignats.
- **idDispositiu (opcional):** identificador del dispositiu del qual es vol obtenir la última dada.

Si no s'especifica ni el paràmetre `idUsuari` ni `idDispositiu`, la petició donarà lloc a un error amb codi 400. Si s'especifica el paràmetre `idDispositiu`, es filtrarà per aquest encara que s'especifiqui el paràmetre `idUsuari`. **Els paràmetres de la petició són OPCIONALS, però és OBLIGATORI especificar-ne almenys un.**

4.9.2 Format de la resposta

```
{
  "success" = True,
  "dades":
  [
    {
      "idDispositiu": (int),
      "nomDispositiu": (str),
      "dataHora": (datetime),
      "dadaHum": (decimal),
      "dadaTemp": (decimal)
    },
    ...
  ]
}
```

4.9.3 Exemples de peticions

- Obtenir la última dada d'un dispositiu:
GET /obtenirUltimaDadaDispositiu?idDispositiu=1
- Obtenir la última dada de tots els dispositius assignats a un usuari:
GET /obtenirUltimaDadaDispositiu?idUsuari=1

4.10 /obtenirModificacionsLlindars

Endpoint de flask per obtenir la última modificació en els llindars de reg de dispositius. Si es demana d'un dispositiu en concret, com que ja s'ha comunicat el canvi en el llindar, es borra de la llista.

4.10.1 Paràmetres de la petició

- **idHardcode (opcional)**: identificador únic del dispositiu del qual es vol obtenir la última modificació en els llindars de reg..

Si no s'especifica el paràmetre, s'obtindran totes les últimes modificacions de llindars de qualsevol dispositiu.

4.10.2 Format de la resposta

```
{
  "success" = True,
  "dades":
  [
    {
```

```
        "idHardcode": (str),
        "llindarMinimReg": (decimal),
        "llindarMaximReg": (decimal)
    },
    ...
]
```

Si es retorna una llista buida, vol dir que no existeix cap modificació dels llindars per als paràmetres especificats.

4.10.3 Exemples de peticions

- Obtenir les últimes modificacions de llindars de tots els dispositius:
GET /obtenirModificacionsLlindars
- Obtenir la última modificació de llindars d'un dispositiu en concret:
GET /obtenirModificacionsLlindars?idHardcode=A