



Gradu amaierako lana / Trabajo fin de grado  
Ingenieritza Elektronikoko gradua / Grado en Ingeniería Electrónica

# Tentsio ertaineko sentsore transformadoreen laborategiaren automatizazioa

Egilea/ Autor/a:

Amaia Anuzita Ziarreta

Zuzendariak/Directores/as:

Raquel Justo Blanco

José M. Alcaide



# Aurkibidea

|  |            |
|--|------------|
| <b>1. Sarrera eta helburuak</b>                          | <b>1</b>   |
| <b>2. Arteche taldea</b>                                 | <b>3</b>   |
| <b>3. Tentsio erdiko laborategia</b>                     | <b>5</b>   |
| 3.1. Entseatu beharreko ekipoa                           | 5          |
| 3.1.1. Low Power Instrument Transformers (LPIT)          | 5          |
| 3.1.2. Power Line Communications (PLC)                   | 8          |
| 3.2. Saiakuntzak   | 9          |
| 3.2.1. Dielektrikoen eta deskarga partzialen saiakuntzak | 9          |
| 3.2.2. Doitasun entsegua                                 | 10         |
| 3.3. Saiakuntzetarako ekipoa                             | 12         |
| 3.3.1. Tentsio igogailua                                 | 12         |
| 3.3.2. Doitasun neurgailua                               | 13         |
| 3.3.3. Deskarga partzialen neurgailua                    | 13         |
| <b>4. Laborategiaren automatizazioa</b>                  | <b>15</b>  |
| 4.1. AIDA proiektua                                      | 15         |
| 4.2. Proiektuaren bizitza zikloa                         | 16         |
| 4.3. Aplikazioa  | 16         |
| 4.3.1. Komunikazio moduluak                              | 18         |
| 4.3.2. Estatu makina                                     | 21         |
| 4.3.3. Interfaze grafikoa                                | 22         |
| 4.3.4. Programa nagusia                                  | 24         |
| 4.4. Bertsioen kontrola                                  | 29         |
| <b>5. Ondorioak</b>                                      | <b>31</b>  |
| <b>A. PuenteZera Klasea</b>                              | <b>34</b>  |
| <b>B. Hipotronics Klasea</b>                             | <b>41</b>  |
| <b>C. ComunicacionWebService Klasea</b>                  | <b>54</b>  |
| <b>D. MaquinaEstados Klasea</b>                          | <b>72</b>  |
| <b>E. App Klasea</b>                                     | <b>76</b>  |
| <b>F. VentanaLogIn Klasea</b>                            | <b>142</b> |
| <b>G. Fluxu-diagramak</b>                                | <b>145</b> |

## 1. Sarrera eta helburuak

Gaur egun, sare-elektrikoaren kudeaketa aldaketa ugari jasaten ari da sortutako baliabideen hazkuntzaren, zerbitzu kalitate baldintza zorrotzagoen eta kotxe elektrikoak bezalako karga berrien ondorioz. Hau dela eta, errendimendu handiko monitorizazioa, automatizazioa eta sarearen ertzetik hurbilago dagoen urruneko kontrola behar dira, nodo berrietara iristeko eta sistemaren operadoreentzako erronka berriak sortuz.

Distribuzio sareko sorgailuetan, funtzio kritiko hauek erdi-tentsioko sentsoreetan oinarritzen dira, hauek funtsezko osagaiak izanik tentsio eta korronteen *RMS* balioen neurketa fidagarriak lortzeko. Sentsoreen erabilerari esker, zerbitzu publikoetako enpresek datu prozesagarriak eskuratzen dituzte planifikazioa hobetzeko, arazoei era eraginkorragoan aurre egiteko, distribuzio sareak integratzeko eta sistemaren mantentzea optimizatzen.

Momentu honetan, Artecheren 54.000 *potentzia baxuko neurketa transformadore* (LPIT) edo sentsore eta akoplagailu baino gehiago daude mundu osoan zehar instalatuta [1] eta merkatuaren joerari erreparaturik, kantitate honek gora baino ez du egingo. Hori dela eta, guztiz beharrezkoa bilakatu da gailu hauen bai fabrikazio eta bai saiakuntza prozesuak transformatzea azkarragoak, fidagarriagoak eta errentagarriagoak izan daitezten.

Honen harira, Artecheko taldeak *Industria 4.0* filosofia korporatiboa jarraitzea erabaki du. Filosofia honen oinarriak ondokoak dira: digitalizazioa, teknologia berrien erabilera, LEAN inizatibak eta prozesu industrialen automatizazioa [2]. Beraz, merkatuaren beharrezkoen erreparaturik eta filosofia berri hau jarraitzeko erabakia behin hartuta laborategi ezberdinak automatizatzeko prozesuari hasiera eman zaio, tentsio ertaineko laborategiak barne.

Ildo beretik, proiektu honen helburu nagusia automatizazio hori ahalbidetzen duen aplikazio bat garatzea da. Horretarako beharrezkoa izan da entseatzeko diren ekipoen ezagutza minimo bat edukitzea, baita burutzen diren entseguen eta erabiltzen diren ekipoen ulertze sakon bat ere. Horregatik, lan honen lehen pausua ikerketa hori burutzea izan da.

Behin informazio guztia biltzean automatizazioaren erdigune den aplikazioa garatu da *Python* programazio lengoaiarekin. Programa honen jomugak ondokoak izan dira:

- Komunikazio eraginkorra laborategiko ekipoekin.
- Informazio jario zuzen eta segurua Artecheko barneko sistemekin.
- Estatu makinan oinarritutako kontrol moduluen ulertzea eta inplementazioa.
- Erabiltzailearentzat ulerterraza den interfaze grafiko baten sorrera.
- Bertsioen kontrol zuzen eta eraginkorra.
- Artecheko barne estandarizazioak jarraitzea bai programazioari, bai interfaze estaloiari dagokionez.

Aipatzekoa da ere proiektu honen helburuen barnean aurkitzen dela lan honen integrazioa Artecheren AIDA (Artecheko Industrial Data Aquisizio) proiektuan. Proiektu hau, lan egiteko eta programa berriak eraikitzeko modu batean oinarritzen da. Honen helburua, programa horiek Artecheko kontrolatutako hardware dispositiboetan exekutatzeko da, makina eta automata industrialekin era segurua eta estandarizatua konektatzea ahalbidetuz.



Honi esker, segurtasunari eta kontrolari dagokien jomugak lortzen dira programaren barnean, baita estandarizazioari dagokienak ere, hauek ezinbestekoak izanik enpresa batean. Izan ere, gaur egun edozein aplikazio elektronikoa garatzean zibersegurtasuna buruan edukitzea beharrezkoa da eta kasu honetan Artecheko taldeak *ISO 27000* zibersegurtasun araua jarraitzeko erabakia hartu du, AIDA proiektua garatuz.

Esan beharra dago, AIDA proiektuari programa jada garatuta zegoen puntu batean eman zitzaiola hasiera. Ondorioz, aplikazioa bere barnean integratu ahal izateko, aldaketa ugari egin behar izan zaizkio programari eta lan egiteko era aldatu behar izan da, batez ere bertsioen kontrolari dagokionez.

## 2. Artech e taldea

Artech e izaera internazionalako enpresa da, instalazio produktibo zein tekniko-komertzialak ditu mundu osoan zehar banatuta eta bere produktuak *175 herrialde* baino gehiagotan aurki ditzakegu. *2.400 langile* baino gehiago ditu eta aktiboki parte hartzen du gaur egungo organizazio elektriko nagusienetan: IEC (International Electrotechnical Commission), IEEE (Institute of Electrical and Electronics Engineers), CIGRE (International Council on Large Electric Systems) eta CIRED (International Conference on Electricity Distribution) [3]. Gainera, ondoko esparruetarako soluzioak garatzen ditu:

- Tentsio altu zein erdiko neurketa transformadoreak
- Neurketa digitala
- Potentzia baxuko transformadoreak eta sentsoreak
- Sare eta sistema elektrikoen automatizazioa
- Errele osagarriak eta proba blokeak
- Tentsio erdiko ezargailua
- Energia kalitatea
- Balio erantsiko zerbitzu espezializatua
- *Turnkey soluzioak* Mexiko eta Brasilen. Horrela deitzen zaie oso erraz implementatu daitezkeen soluzioei, enpresak eskura dituen errekurtsuak soilik baliatuz.

Taldea *20 enpresa* baino gehiagoz osatuta dago eta horietatik Espainian kokatuta hurrengoak daude:

- Artech e Centro de Tecnología (ACT)
- Electrotécnica Artech e Hermanos (EAHSL)
- Artech e Gas Insulated Transformers (AGIT)
- Electrotécnica Artech e Smart Grid (EASG)
- Sistemas Avanzados de Control (SAC)

Mundu mailako instrumentazio transformadoreen fabrikatzaile nagusienetariko bat da eta proiektu honi dagokionez, *10 urte* baino gehiago daramatza tentsio erdiko sentsoreetan eta automatizazioan lanean. Horrez gain, transformazio zentroen monitorizazio eta automatizazio prozesuetan sentsore, PLC akoplagailu eta IED (Intelligent Electronic Device) hornitzaile nagusia izan zen.

Gaur egungo sentsoreen produktu lerroak tentsio zein korrante sentsoreak ditu, baita bien konbinaketa ere. Horrez gain, barneko zein aire libreko aplikazioetarako produktuak garatzen ditu. Artecheren potentzia baxuko instrumentazio transformadore eta sentsoreek, adimendun dispositibo elektroniko modernoekin konbinatuta, sistemaren ulermen hobea ahalbidetzen dute eta sarearen funtzionamenduaren gestioa optimizatzen dute.

Sentsoreen produktu gama honek ematen dituen abantaila nagusiak ondokoak dira [1]:

- **Estandarizazioa:** erreferentzia bakar batek tentsio maila zabal eta aplikazio ugari betetzen ditu.

- Ekipo konbentzionalekin **trukakortasuna**: ez dute kalibraziorik behar *in situ*.
- **Segurtasuna**: erabiltzailea ez dago kontaktuan tentsio maila arriskutsuekin, izan ere, irteera sekundarioak potentzia baxukoak dira. Are gehiago, neurketa ekipo konbentzionaletan ez bezala, sekundarioa irekita uzteak ez dakar inolako arriskurik.
- Ekipoen **trinkotasuna** eta **arintasuna**: instalazio aukera berriak.
- **Doitasuna**: *IEC-61869* araua jarraituz, *0.5P* klaseko sentsoreak.

## 3. Tentsio erdiko laborategia

### 3.1. Entseatu beharreko ekipiak

Laborategian bai muntatzen eta bai probatzen diren tresnak bi talde nagusitan banatzen dira: *sentsoreak* eta *PLC akoplagailuak*. Bereizmen hau sare elektrikoaren barnean duten funtzionalitateagatik egiten da, gero fabrikazio prozesu oso antzekoa duten arren.

Alde batetik, sentsoreen funtzio nagusia sare elektrikoaren akatsak antzematea da. Hauek neurtzen duten seinalea babes ekipotara konektatzen da zuzenean eta horri esker sarearen mantentze lan eraginkorrago bat burutzen da: akatsa antzeman, kaltetutako zatia isolatu, konpondu eta sarea berrezarri. Horrez gain, karga fluxuen kontrola ere burutzen dute sareko galerak edota lapurretak antzematea ahalbidetuz. Aipatzekoa da ekoizpen banatuaren azterketaz arduratzen direla.

Bestetik, PLC akoplagailuek eroale batean zehar datuak garraiatzea ahalbidetzen dute, tentsio erdiko AC korrontearen distribuzioa burutzen duten bitartean. Honi esker, banda zabaleko IP sare bat lortzen da jada existitzen diren sare elektrikoaren gainean.

#### 3.1.1. Low Power Instrument Transformers (LPIT)

*Potentzia baxuko tentsio transformadoreak* (LPIT), neurketa transformadore ez konbentzional edo sentsore ere deituak, potentzia baxuko irteerako seinale analogiko edo digital bat ematen du, korrontean zein tentsioan. Seinale hau neurketa, babes edo antzeko ekipotara bideratzen da. Bi mota daude [4]:

- **LPCT** : potentzia baxuko korronte transformadoreak. *Korrontea* neurtzen dute.
- **LPVT** : potentzia baxuko tentsio transformadoreak. *Tentsioa* neurtzen dute.



**Irudia 1:** ezkerrean LPVT OVERSENS modelo, eskuman LPIT bat [1]

Biek eskaintzen dute irteerako seinale bat mV magnitude-ordenakoa. Sentsore hauek neurketa transformadore konbentzionalen funtzionalitate berbera eskaintzen dute abantaila batzuekin:

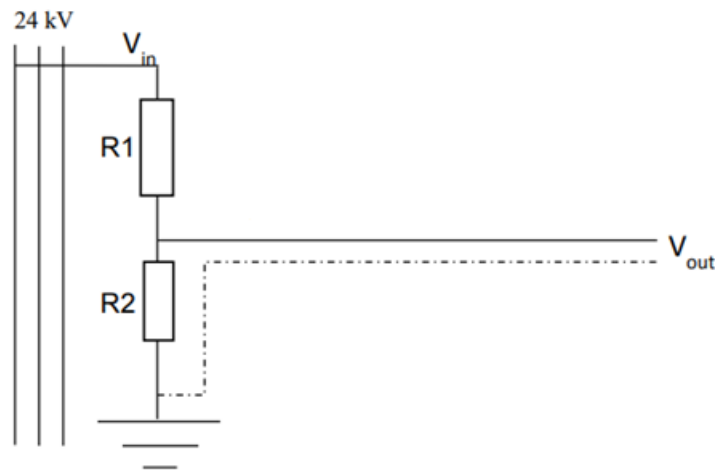
- Tamaina eta pisu txikiagoak.
- Linealtasun ona.
- Tentsio baxuko irteerako seinalea (mV). Honi esker ekipoen eta pertsonalaren segurtasuna nabarmen handitzen da.

Laborategi konkretu hauetan LPVT-ak soilik entseatzeko direnez, hauen azalpen sakonagoan zentratuko naiz, LPCT-ak alde batera utziz.

### LPVT (Potentzia baxuko tentsio transformadoreak)

Tentsio tentsioak nukleo magnetiko gabeko transformadoreak dira, zirkuitu tentsio zatitzaile batean oinarrituak. Helburua neurtu nahi den parametroa murriztea da potentzia baxuko irteera bat emanez. Hau bi elementu erresistikor edo kapazitibo seriean jarritz lortzen da.

Sentsore hauek elementu pasiboz soilik daude osaturik eta horri esker ez da elikadura iturririk behar. Eskema elektrikoari dagokionez, Artechek eskaintzen dituen tentsio sentsore guztiak tentsio zatitzaile erresistikorrak dira Irudia 2-n ageri den bezala. Bi erresistentziaz osatuta daude, seriean kokatuta, sarrerako tentsioa irteerako tentsio txikiago batera bihurtzeko. Konfigurazio honetan, erresistentzia primarioa sekundarioa baino askoz ere handiago da eta honi esker lortzen da mV magnitude-ordenako seinalea.



**Irudia 2:** *tentsio zatitzaile erresistikorraren eskema elektriko.*

Transformazio erlazioa, 1 Ekuazioan ageri dena, sentsorearen sarrera eta irteerako tentsioen arteko erlazioa izango da. Hau ezaguna izanik eta sentsorearen irteera sekundarioko seinalea ezaguturik, sareko tentsioa kalkulatu daiteke edozein momentutan.

$$I = \frac{V_{in}}{R_1 + R_2} = \frac{V_{out}}{R_2} \quad (1)$$

Sentsore eta beste neurketa transformadore guztiek jarraitzen duten IEC araua Irudia 3-n azaltzen da. LPVT-en arau espezifikoak *IEC 61869-11* [5] erreferentzian biltzen da, *61869-1* estandarra (neurketa transformadoreak) eta *61869-6* (LPIT) osatuz.

Sentsoreen ezaugarri elektriko nagusienetariko bat doitasuna da. Ezaugarri hau neurketa errorearekin erlazionatuta dago eta bi errore ezberdinen konbinaketa da:

- **Erlazio errorea:** transformadoreak neurketan sartzen duen errorea. Hau sortzen da transformazio erlazio erreala ez delako transformazio erlazio nominalaren berdina
- **Desplazamendua fasean edo desfasea:** fase diferentzia tentsio primario eta sekundarioaren fasearen artean. Hau zero izango da transformadore ideal baten kasuan soilik.

| PRODUCT FAMILY STANDARDS                                    | PRODUCT STANDARD  | PRODUCTS   | OLD STANDARD  |         |
|---|---|--|---|---------|
| 61869-1<br>GENERAL REQUIREMENTS FOR INSTRUMENT TRANSFORMERS | 61869-2   | ADDITIONAL REQUIREMENTS FOR CURRENT TRANSFORMERS           | 60044-1<br>60044-6  |         |
|   | 61869-3   | ADDITIONAL REQUIREMENTS FOR INDUCTIVE VOLTAGE TRANSFORMERS | 60044-2   |         |
|   | 61869-4   | ADDITIONAL REQUIREMENTS FOR COMBINED TRANSFORMERS          | 60044-3   |         |
|   | 61869-5   | ADDITIONAL REQUIREMENTS FOR CAPACITOR VOLTAGE TRANSFORMERS | 60044-5   |         |
|   | 61869-6<br>ADDITIONAL GENERAL REQUIREMENT FOR LOW-POWER INSTRUMENT TRANSFORMERS | 61869-7  | ADDITIONAL REQUIREMENTS FOR ELECTRONIC VOLTAGE TRANSFORMERS   | 60044-7 |
|   |   | 61869-8  | ADDITIONAL REQUIREMENTS FOR ELECTRONIC CURRENT TRANSFORMERS   | 60044-8 |
|   |   | 61869-9  | DIGITAL INTERFACE FOR INSTRUMENT TRANSFORMERS   |         |
|   |   | 61869-10   | ADDITIONAL REQUIREMENTS FOR LOW-POWER PASSIVE CURRENT TRANSFORMERS                                      |         |
|   |   | 61869-11   | ADDITIONAL REQUIREMENTS FOR LOW-POWER PASSIVE VOLTAGE TRANSFORMERS                                      | 60044-7 |
|   |   | 61869-12   | ADDITIONAL REQUIREMENTS FOR COMBINED ELECTRONIC INSTRUMENT TRANSFORMER OR COMBINED PASSIVE TRANSFORMERS |         |
|   |   | 61869-13   | STAND ALONE MERGING UNIT  |         |
|   |   | 61869-14   | ADDITIONAL REQUIREMENTS FOR CURRENT TRANSFORMERS FOR DC APPLICATIONS                                    |         |
|   |   | 61869-15   | ADDITIONAL REQUIREMENTS FOR DC VOLTAGE TRANSFORMERS FOR DC APPLICATIONS                                 |         |

**Irudia 3:** *sentsoreentzako IEC araua [5].*

Bai erlazio errorea bai desfasea kontuan hartuta, doitasun klase bat ezartzen zaio sentso-reari. Honek erlazio errorea eta desfasea limite espezifiko batzuen barruan finkatzen ditu. Neurketa erroreari eta ondorioz doitasunari eragiten dioten faktoreen artean hauek dira garrantzitsuenak:

- **Temperaturaren egonkortasuna:** sentsorearen irteerako balioek temperaturarekin aldatzeko joera daukate.
- **Tentsioaren egonkortasuna:** tentsioaren bariazioek aldaketak eragiten dituzte neurketetan.
- **Egonkortasuna denboran:** luzaroko esposizio batek bai muturreko temperature-tara eta bai tentsio altuetara ekipoen degradazioa ekar dezake.

Segurtasun eta neurketa aplikazioetarako erroren limiteak doitasun klase bakoitzarentzako Irudia 4-en biltzen dira. Artecheren kasuan, gaur egun 0.5P klaserainoko tentsio sentsoreak aurki ditzakegu bai neurketarako eta bai segurtasunerako erabiliak.

Ildo beretik, doitasunaz gain tentsio sentsoreak definitzen dituzten ezaugarri nagusiak ondokoak dira, hauek ere *IEC- 61869-11 [5]* arauak ezarriak:

- **Tentsio onargarri maximoa (Um):** ekipiak etengabe lanean jasan dezakeen tentsioaren balio efikaz altuena bi faseen artean. Adibidez, 24kV.
- **Erresistentzia dielektrikoa:** ekipiak minutu batean zehar jasan dezakeen tentsioaren balio efikaz altuena. Adibidez, 50kV.
- **Tximista bulkada:** ekipiak jasan dezakeen tentsio mutur handiena. Adibidez,

| Accuracy class | Ratio error $\varepsilon$ , $\varepsilon_{cor U}$<br>$\pm$ % |     |     |     |               | Phase error $\varphi_e$ , $\varphi_{cor \varphi_0}$ |     |     |     |               |                         |     |      |      |               |
|----------------|--|-----|-----|-----|---------------|---|-----|-----|-----|---------------|-------------------------|-----|------|------|---------------|
|                |  |     |     |     |               | at voltage (% of rated)                             |     |     |     |               | $\pm$ minutes           |     |      |      |               |
|                | at voltage (% of rated)                                      |     |     |     |               |   |     |     |     |               | at voltage (% of rated) |     |      |      |               |
|                | 2  | 20  | 80  | 100 | $F_v$<br>x100 | 2   | 20  | 80  | 100 | $F_v$<br>x100 | 2                       | 20  | 80   | 100  | $F_v$<br>x100 |
| 0,1P           | 0,5  | 0,2 | 0,1 | 0,1 | 0,1           | 20  | 10  | 5   | 5   | 5             | 0,6                     | 0,3 | 0,15 | 0,15 | 0,15          |
| 0,2P           | 1  | 0,4 | 0,2 | 0,2 | 0,2           | 40  | 20  | 10  | 10  | 10            | 1,2                     | 0,6 | 0,3  | 0,3  | 0,3           |
| 0,5P           | 2  | 1   | 0,5 | 0,5 | 0,5           | 80  | 40  | 20  | 20  | 20            | 2,4                     | 1,2 | 0,6  | 0,6  | 0,6           |
| 1P             | 4  | 2   | 1   | 1   | 1             | 160   | 80  | 40  | 40  | 40            | 4,8                     | 2,4 | 1,2  | 1,2  | 1,2           |
| 3P             | 6  | 3   | 3   | 3   | 3             | 240   | 120 | 120 | 120 | 120           | 7                       | 3,5 | 3,5  | 3,5  | 3,5           |
| 6P             | 12   | 6   | 6   | 6   | 6             | 480   | 240 | 240 | 240 | 240           | 14                      | 7   | 7    | 7    | 7             |

The accuracy class shall be defined in accordance with the relay manufacturer's sizing rules and application requirements. LPVT accuracy class 0,1P could be recommended for power systems earthed with Petersen coil while LPVT with accuracy classes 0,2P or 0,5P could be used for other earthing systems.

**Irudia 4:** doitasun klaseak eta bakoitzari dagozkion errore limiteak [5].

125kV.

- **Deskarga partziala:** baldintza hau 7,2 kV edo gehiagoko tentsio onargarri maximoa ( $U_m$ ) duten transformadoreei soilik ezartzen zaie. Ezarritako limiteak Taula 1-n ematen dira:

| Tentsioa           | Balio maximoa |
|--------------------|---------------|
| 1,2 $U_m$          | 50 pC         |
| 1,2 $U_m/\sqrt{3}$ | 20 pC         |

**Taula 1:** Deskarga partzialen limiteak

- **Transformazio erlazio nominala:** tentsio primario nominalaren eta tentsio sekundario nominalaren arteko erlazioa. *IEC 61869-11* arauaren arabera, sekundarioaren tentsioaren balio estandarra ( $V_{out}$ )  $3,25/\sqrt{3}$  V da (primarioaren tentsioaren balioarekiko independente). Hala ere, 10.000/1 tentsio transformazio erlazioa ere aztertzen da. Adibidez,  $20.000/\sqrt{3}$  balioko tentsio primario ( $V_{in}$ ) batentzat,  $2/\sqrt{3}$  balioko tentsio sekundario ( $V_{out}$ ) bat.
- **Tentsio faktorea:** faktore biderkatzailea, (1,2 – 1,5 – 1,9), tentsio primario nominalari aplikatzen zaiona tentsio maximoaren balio bat zehazteko. Balio maximo honetan transformadoreak baldintza termiko nabarmenak bete behar ditu ezarritako denboran zehar (jarraituan – 30s – 8h), baita dagozkion doitasun ezaugarriak ere.
- **Karga nominala:** kargaren balioa non doitasun ezaugarriak betetzen diren. Karga honek sentsorea gero konektatuko den ekipoa inpedantzia irudikatzen du. Arauak biltzen duen balio estandarra  $2M\Omega$  da.
- **Frekuentzia nominala:** frekuentzia non ekipoa ezaugarriak definitu diren.

### 3.1.2. Power Line Communications (PLC)

Atal honetan sentsoreak alde batera utziz zelula berdinean bai muntatzen eta bai saiatzten diren beste dispositiboan azalpen labur bat egingo dut: PLC akoplagailuak. Hauek ez

dira neurketetan erabiltzen, komunikazioan baizik.



**Irudia 5:** PLC akoplagailua, comART PLUGCAP modeloa [1].

*Power Line Communications* teknologiak tentsio baxu zein erdiko sare elektrikoak banda zabaleko telekomunikazio sare bezala erabiltzea ahalbidetzen du. Honi esker, frekuentzia altuko seinaleak igorri daitezke sareetako potentzia eroaleen bitartez. Modu honetan, akoplagailu bat jarritz sarearen mutur bakoitzean, komunikazioa burutu daiteke infraestrukturatzat sare elektriko baliatuz. Bi mota daude:

- **Akoplagailu kapazitiboak:** akoplagailua kontaktu zuzenean dago tentsioarekin eta konektagarria edo airerako isolamendukoa izan daiteke.
- **Akoplagailu induktiboak:** tentsio baxuko akoplagailu bat da, isolatutako kablea besarkatzen duena eta seinalea indukzio bidez injektatzen duena tentsio erdiko kablean.

## 3.2. Saiakuntzak

Behin ekipoen fabrikazioa bukatzen denean, nahitaezkoak diren saiakuntzak burutzen dira lortutako balioak dagokien IEC arauari doitzen direla baieztatzeko, kasu honetan *IEC 61869* arau familia.

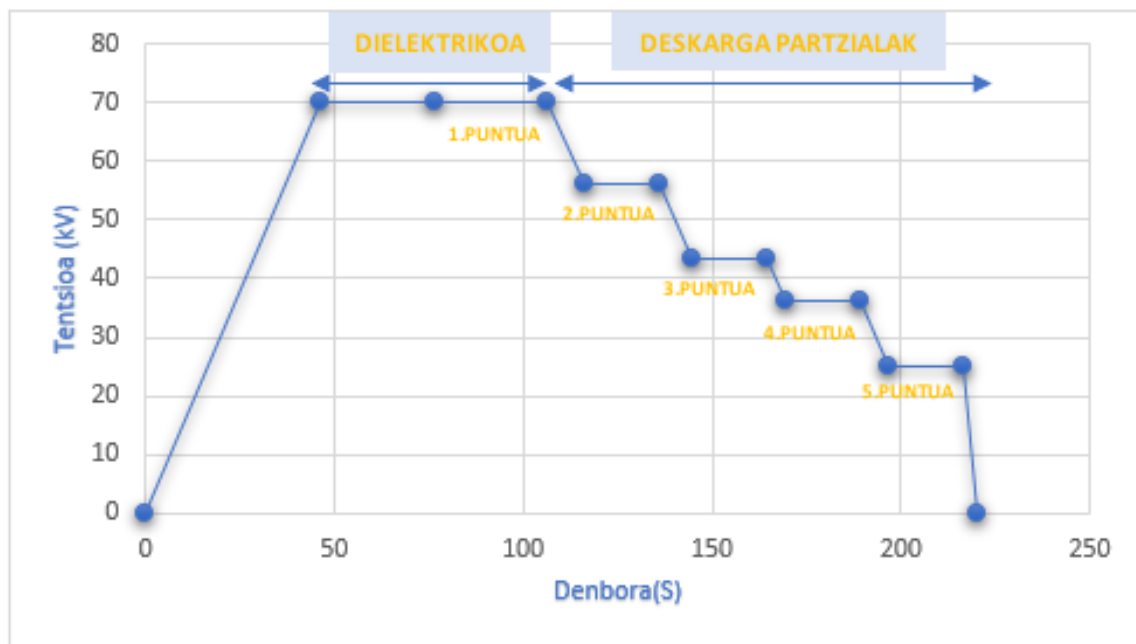
Lau entsegu mota bereizten dira: ikusmen-inspekzioa, entsegu dielektrikoa, deskarga partzialen saiakuntza eta doitasunaren saiakuntza. Guri interesatzen zaizkigunak azken hirurak dira eta batez ere doitasunaren saiakuntza hori izan baita guztiz automatizatzea lortu dena. Aipatzeko da ere sentsoeren kasuan entsegu denak burutzen direla baina PLC akoplagailuei, neurketetarako erabiltzen ez direnez, doitasun entsegua ez zaiela egiten.

### 3.2.1. Dielektrikoen eta deskarga partzialen saiakuntzak

Dielektrikoen eta deskarga partzialen saiakuntzak aldi berean burutzen dira. Lehenengoaren helburua isolamendua hondatzen ez dela egiaztatzea da eta ondorioz, ekipoa ezarriko zaizkion ingurune ezberdinak jasateko gai izango dela bermatzen da. Bigarrenarena aldiz, isolamendu horrek eduki ditzakeen akatsak antzematea, ebaluatzea eta lokalizatzea da. Izan ere, isolamendua galtzeak deskarga elektriko txiki bat sortzen du eta hori da entsegu honetan neurtzen dena.

Dielektrikoen kasuan tentsio nominala baino altuagoa den *tentsio espezifiko bat* aplikatzen da primarioaren eta sekundarioaren artean, *30 segundotan* zehar *50Hz-tara*. Behin denbora hau pasatzean beste 30s mantendu behar da tentsio balio hori balioztatzeko. Deskarga partzialentzako, Grafikoa 1-en ikus daitekeen moduan, neurketak dielektrikoaren entsegua burutu ostean egiten dira tentsioa jaisten doan heinean. Neurketak 5 puntu ezberdinetan burutzen dira, tentsioa konstante mantentzen den puntuetan hain zuzen ere.





**Grafikoa 1:** BARSSENS sentsore modeloaren dielektriko eta deskarga partzialen entseguak burutzeko erabiltzen den arrapala tentsio seinalea denborarekiko.

Entsegu dielektrikoa ontzat ematen da isolamenduan zulaketarik edo deformaziorik eragiten ez denean. Deskarga partzialak picocoulombetan neurtzen dira eta neurtutako balioak arauak ezarritako balio maximoak baino txikiagoak izan behar dira. Baldintza hauek betetzen ez badira, ekipoa baztertu egiten da. Balio horien adibide bat Taula 2-n ikus daiteke.

| Puntuak | Tentsioa (kV) | Deskarga partzialen limitea (pC) |
|---------|---------------|----------------------------------|
| 1       | 70            | 0                                |
| 2       | 56            | 0                                |
| 3       | 43.2          | 50                               |
| 4       | 36            | 0                                |
| 5       | 24.9          | 20                               |

**Taula 2:** Deskarga partzialen puntuak tentsio eta limiteak PLUGSENS-36 sentsore modeloarentzat.

Limiteak 3 eta 5 puntuetan soilik ezartzen dira. Beste balioak estudio estatistikoak egiteko erabiltzen dira.

### 3.2.2. Doitasun entsegua

Doitasun entsegua giro tenperaturan eta karga nominalera burutzen da. Bertan, tentsio neurketaren errorea neurtzen da, tentsio hein jakin batentzat. Horrez gain, erroreek IEC arauak ezarritako balioa gainditzen ez dituztela egiaztatzen da. Aurretik aipatutako doitasun klase bakoitzak erroreen limite espezifiko batzuk dauzka. Entsegu hau fabrikatzen diren sentsore guztiei egiten zaie.

Sentsore bakoitzari eskatzen zaion doitasun klasea bere aplikazioaren arabera da eta kasu bakoitzean neurketak tentsio ezberdinetan egin behar dira. Hau Taula 3-n laburbiltzen da. Arteheren produktu gamaren barnean tentsio sentsore guztiak  $0.5P$  eta  $1P$  multifuntzio sentsoreak dira.

| Aplikazioa                             | Doitasun klasea | Neurketa tentsioa |
|--|-----------------|-------------------|
| Neurketak                              | 0.1             |                   |
|  | 0.2             | %80               |
|  | 0.5             | %100              |
|  | 1               | %120              |
|  | 3               |                   |
| Neurketak + segurtasuna (multifuntzio) | 0.1P            | %2                |
|  | 0.2P            | %20               |
|  | 0.5P            | %80               |
|  |                 | %100              |
| Segurtasuna                            |                 | %100xFv*          |
|  | 1P              | %2                |
|  | 3P              | %20               |
|  | 6P              | %80               |
|  |                 | %100              |
|  |                 | %100xFv*          |

\*Tentsio faktorea. Segurtasun sentsoreen kasuan 1.9 izan ohi da.

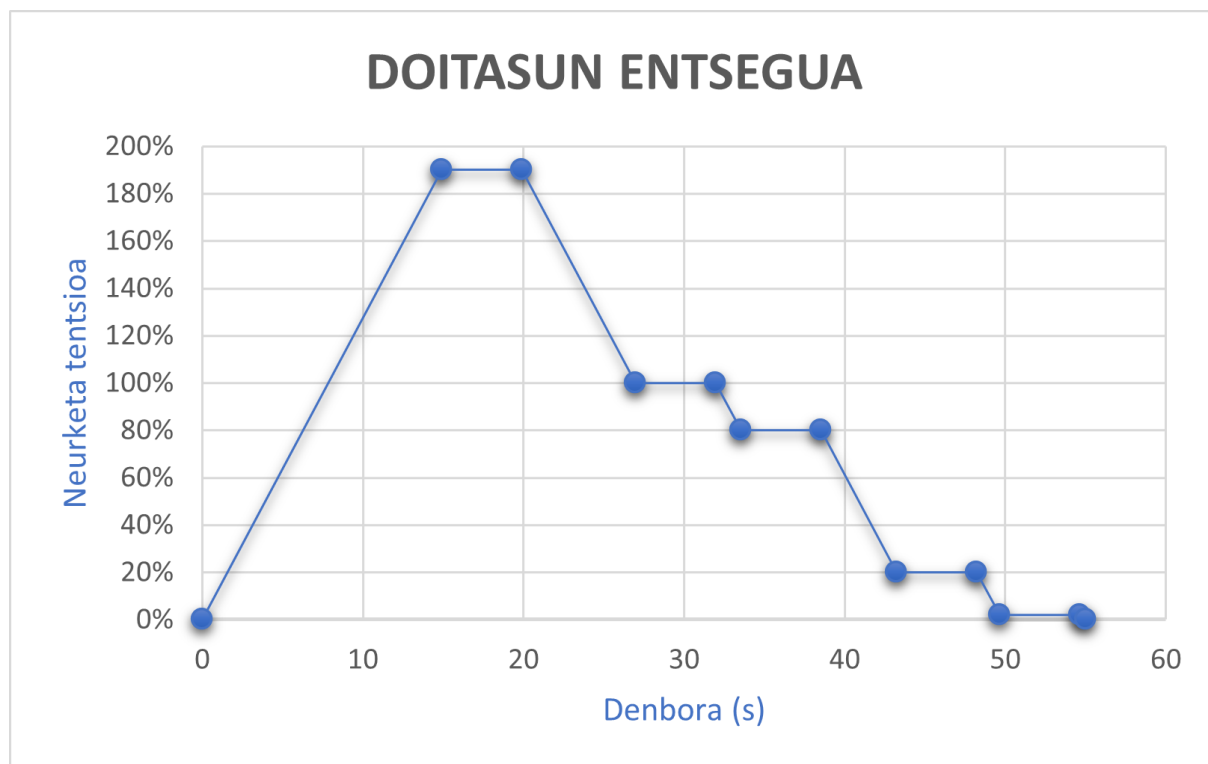
**Taula 3:** doitasun klaseak eta bakoitzean neurtu beharreko tentsioen balioak tentsio nominalarekiko (%) sentsore aplikazio ezberdinentzako [5]

| Accuracy class | Ratio error $\delta$ , $\delta_{cor U}$ |     |     |     |               | Phase error $\phi_e$ , $\phi_{cor \phi_0}$ |     |     |     |               |                         |     |      |      |               |
|----------------|---|-----|-----|-----|---------------|--|-----|-----|-----|---------------|-------------------------|-----|------|------|---------------|
|                | $\pm$ %                                 |     |     |     |               | $\pm$ minutes                              |     |     |     |               | $\pm$ centiradians      |     |      |      |               |
|                | at voltage (% of rated)                 |     |     |     |               | at voltage (% of rated)                    |     |     |     |               | at voltage (% of rated) |     |      |      |               |
|                | 2                                       | 20  | 80  | 100 | $F_v$<br>x100 | 2  | 20  | 80  | 100 | $F_v$<br>x100 | 2                       | 20  | 80   | 100  | $F_v$<br>x100 |
| 0,1P           | 0,5                                     | 0,2 | 0,1 | 0,1 | 0,1           | 20   | 10  | 5   | 5   | 5             | 0,6                     | 0,3 | 0,15 | 0,15 | 0,15          |
| 0,2P           | 1                                       | 0,4 | 0,2 | 0,2 | 0,2           | 40   | 20  | 10  | 10  | 10            | 1,2                     | 0,6 | 0,3  | 0,3  | 0,3           |
| 0,5P           | 2                                       | 1   | 0,5 | 0,5 | 0,5           | 80   | 40  | 20  | 20  | 20            | 2,4                     | 1,2 | 0,6  | 0,6  | 0,6           |
| 1P             | 4                                       | 2   | 1   | 1   | 1             | 160  | 80  | 40  | 40  | 40            | 4,8                     | 2,4 | 1,2  | 1,2  | 1,2           |
| 3P             | 6                                       | 3   | 3   | 3   | 3             | 240  | 120 | 120 | 120 | 120           | 7                       | 3,5 | 3,5  | 3,5  | 3,5           |
| 6P             | 12                                      | 6   | 6   | 6   | 6             | 480  | 240 | 240 | 240 | 240           | 14                      | 7   | 7    | 7    | 7             |

The accuracy class shall be defined in accordance with the relay manufacturer's sizing rules and application requirements. LPVT accuracy class 0,1P could be recommended for power systems earthed with Petersen coil while LPVT with accuracy classes 0,2P or 0,5P could be used for other earthing systems.

**Irudia 5:** erroreen balioen limiteak IEC-61869-11[5] arauaren arabera.

Atal honekin bukatzeko, entsegu honentzako IEC araua, erroreen baloreen limiteekin bai modulan eta bai angeluan Irudia 5-en biltzen da. Horrez gain, dielektrikoen eta deskarga partzialen kasuan bezala, tentsio arrapala bat erabiltzen da neurketak egiteko, kasu honetan denbora laburragoetan Grafiko 2-n ikus dezakegun bezala.



**Grafikoa 2:** *OVERSENS-25* sentsore modeloaren doitasun entsegua burutzeko erabiltzen den arrapala tentsio seinalea denborarekiko. Tentsioaren balioa tentsio nominalarekiko adierazita dago. Tentsio nominala: 11.76 kV. Tentsio faktorea: 1.9.

### 3.3. Saiakuntzetarako ekipoak

Aurreko atalean azaldutako saiakuntzak burutu ahal izateko hiru ekipo ezberdin erabiltzen dira: tentsio igogailua, deskarga partzialen neurgailua eta doitasun neurgailua.

#### 3.3.1. Tentsio igogailua

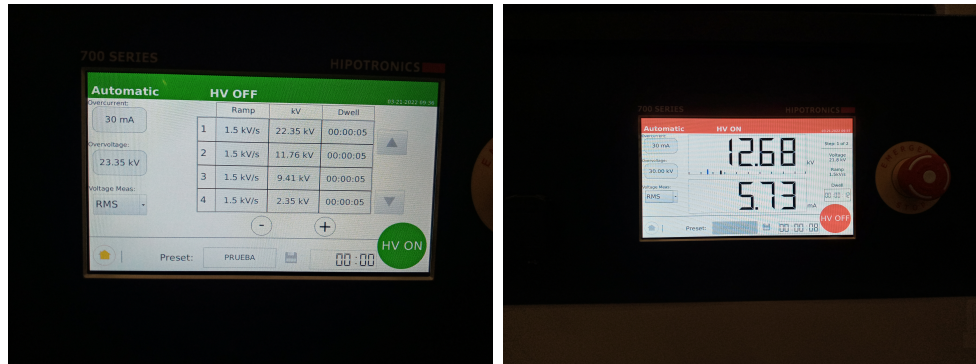
Bai doitasun entseguan bai deskarga partzialen eta dielektrikoen entseguan beharrezkoak diren tentsio arrapalak ematen dituen makina. Artecheko laborategi askotan tentsioa oraindik eskuz (erruleta baten bitartez) igotzen den arren, sentsoreen tentsio erdiko laborategian operazio hau automatizatzen duen *HIPOTRONICS 775-10D5-22740* modelo dago eskuragarri.



**Irudia 6:** *HIPOTRONICS 775-10D5-22740* tentsio igogailua [6].

Sistema integratutako kontrol interfaze digital batek eta tentsio erregulatzailer aldakor batek osatzen dute [6]. Erabiltzaileak interfazearen bitartez eskuzko kontrola edo kontrol

automatikoa aukeratzen du. Modu automatikoan, entsegurako erabiliko den arrapala eta beste parametro batzuk definitzen ditu eta azkenik **HV ON** botoia sakatzen du. Behin prozesua hasten denean momentu oroko tentsioa zein korrrontea pantailan ikusten dira eta gelditzeko **HV OFF** botoia sakatu behar da (Irudia 7).



**Irudia 7:** ezkerrean modu automatikoa aukeratzean agertzen den pantaila. Eskuman HV ON botoia sakatzean agertzen den pantaila.

### 3.3.2. Doitasun neurgailua

Doitasun entsegua burutzeko *ZERA WM3000U* tentsio neurgailu zubia erabiltzen da. Zubia doitasun handiko konparazio-unitatea da. Saiakuntzako transformadorearen bigarren mailako seinalea gailu estandar batek emandako erreferentzia-seinale batekin konparatzen duena [7].

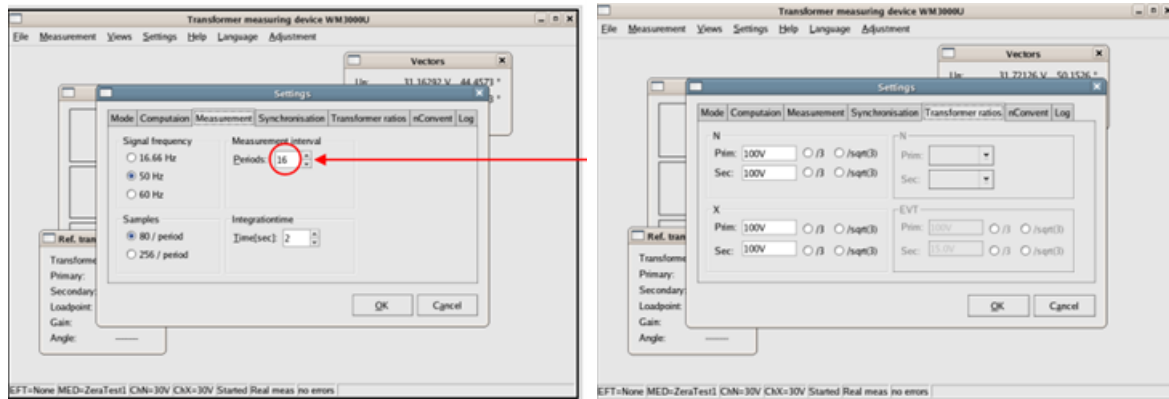


**Irudia 8:** *ZERA WM3000U* tentsio neurgailu zubia [7].

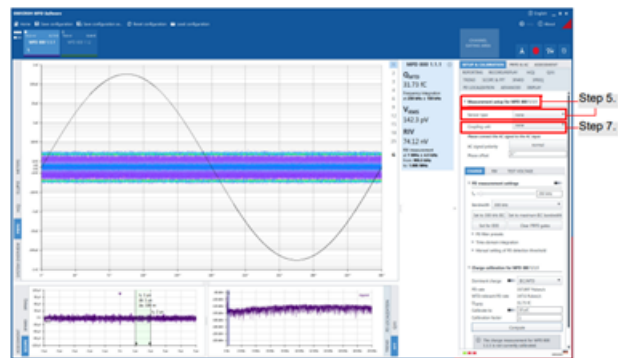
Erlazio zein desplazamendu errorea eta tentsioa pantailan erakusten dira. Horrez gain, konfigurazioa aldatzeko aukera ematen du, baita erreferentziako-seinalearen parametroak eta entseguko parametroak aldatzeko aukera ere (frekuentzia, tentsio erlazioak,...) (Irudia 9).

### 3.3.3. Deskarga partzialen neurgailua

Deskarga partzialak neurtzeko *OMICRON MPD 800* eskuratzeko unitatea eta *OMICRON* enpresaren software berezi bat erabiltzen dira. Honen automatizazioa proiektutik kanpo gelditu denez alde batera utziko dut.



**Irudia 9:** neurketa konfigurazioa eta transformadoreen ratioak aldatzeko pantailak [7].



**Irudia 10:** OMICRON MPD 800 eskuratzeko unitatea eta OMICRON enpresaren softwarea [8].

Aipatzekoa da zelulako langileek Irudia 10-eko grafikoa interpretatzen dutela eta ondoren interpretazio horretatik balioak apuntatzen dituztela gero datu base batean gordetzeko.

## 4. Laborategiaren automatizazioa

Enpresa handi baten barnean prozesu bat automatizatzeko erabakia hartzen den unean proiektu oso bat jartzen da martxan. Hau da, ez da soilik programazio bidezko aplikazio erabilgarri eta funtzional bat sortzea. Konkreteriki Artecheko taldeak protokolo jakin bat espezifikatu du egoera honetarako, AIDA. Aurretik esan bezala, AIDA proiektuari automatizazio proiektua jada hasita zegoenean eman zitzaion hasiera. Horregatik, lan honen barnean dago laborategiaren automatizaziorako aplikazioaren sorrera eta baita honen integrazioa AIDA proiektuaren barnean.

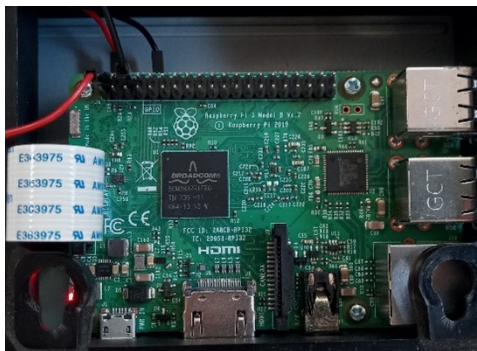
### 4.1. AIDA proiektua

AIDA (Artecheko Industrial Data Aquisition) software batean eta lan egiteko eta programa berriak eraikitze modu batean oinarritzen da. Honen helburua, programa horiek Artecheko kontrolatutako hardware dispositiboetan exekutatzeko da, makina eta automata industrialekin era seguru eta estandarizatuan konektatzeko ahalbidetuz.

Zehazki, proiektuak hiru zutabe ditu:

- Laborategiko makina eta automaten zuzenean konektatzen diren dispositiboak:

Hasiera batean PC bat zen baina momentu honetan **Raspberry PI** dispositiboa da. Raspberry PI-a Artecheko sistema informatikoen ekipoa aurretik konfiguratu eta segurtatu du, zona seguru bihurtuz zibersegurtasun aldetik. Programa, kode unitate ere deituko duguna, dispositibo honetan exekutatu da eta postuko funtzionalitate espezifikoaz deituko du Raspberry PI-a.



**Irudia 11:** *Raspberry PI dispositibo elektronikoa.*

- Software bideratzailea, Artecheko zerbitzari zentrolean instalatuta egongo dena:

Zerbitzu honen lana Raspberry bakoitzean exekutatu diren kode unitateak mantentzea eta kontrolatzea da. Izan ere, proiektu honetarako Raspberry PI bakarra erabiliko da, postu bakarraren automatizazioa delako; baina, Artecheko laborategi ezberdin askotan Raspberry ugari daude martxan, bakoitza bere programa espezifikoarekin.

Software bideratzaileak, interfaze bateratu batetik ondokoa administratzen du: martxan dauden dispositiboak, exekutatu ari diren programen bertsioak, hauen egoera, etab. Nagusiki, dagokion gailuetan instalatutako programa bakoitzaren azken bertsio erabilgarria mantentzeaz arduratzen da.

- Kode unitateak:

Proiektu honek kode unitateei dagokienez ondokoa espezifikatzen du: programazio lengoaia **Python** izatea, interfaze grafikoak egiteko **Qt Designer** erabiltzea eta objektuei zuzendutako programazioa erabiltzea. Honez gain, programaren barnean kontrol hobe bermatzeko estatu makina sortzea eta bertsioen kontrolerako **Gitlab** erabiltzea adierazten du. Bai hau, bai estiloari dagokion beste hainbat espezifikazio Artecheren programaziorako liburu zurian agertzen dira. Honen helburua, ondo funtzionatzen duen programa bat edukitzea da, ahal den heinean homogeneitate bat bermatuz, bai kalitatean eta bai funtzionalitatean.

## 4.2. Proiektuaren bizitza zikloa

Esan bezala, enpresa batean proiektu berri batekin hasteko momentuan prozesu oso bat jartzen da martxan. Kasu honetan aurrera eramandako pausuak ondokoak izan dira:

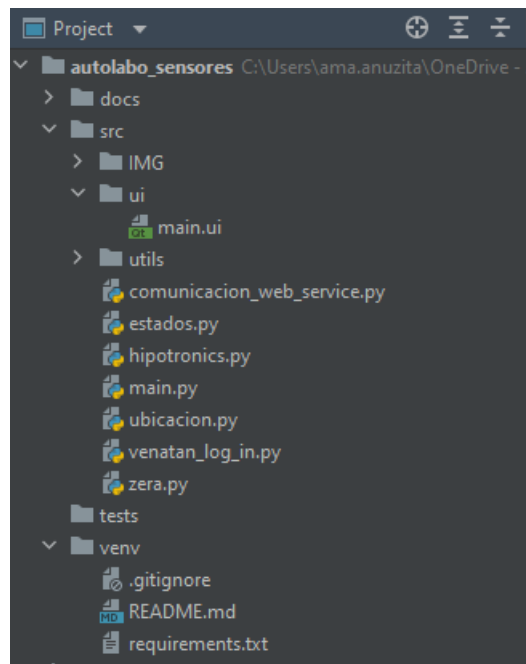
1. Prozesu industrialeko puntu batean automatizazio beharrizan bat detektatu da: tentsio ertaineko sentsoeren entsegu laborategia.
2. Aukeratutako prozesuaren automatizaziorako proiektu bat garatu eta exekutatu da, emaitza moduan *Konputagailua + Makina* erako multzo bat sortuz zein era semi-automatikoa funtzionatzeko gai den. *Konputagailua*: Raspberry PI konputagailua. *Makinak*: tentsio igogailua eta doitasun neurgailua.
3. Software bat deiseinatu eta implementatu da konputagailuan txertatzeko. Honi esker prozesuaren azken automatizazioa burutzen da, Artecheko datu baseetatik datuak erauzten dira eta emaitzak biltzen dira. Softwarea kode unitate ugariz osatuta dago eta *Pythoneko autolabo\_sensores* proiektuaren barnean biltzen dira guztiak. Bere barnean daude bai tentsio igogailu eta bai doitasun neurgailuarekin komunikatzeko moduluak eta web service bitartez datu baseekin lan egiteko moduluak. Horrez gain, interfazei dagozkien bai pythoneko moduluak eta bai **.ui** erako *Qt Designer* bidez sortutako fitxategiak osatzen dute.
4. Kode unitatea bizitza ziklo guztian zehar Artecheko biltegi zentral ofizialean gorde da *Gitlab* erabiliz. Kasu konkretu honetan bertsioen kontrol errazagoa egiteko, *Gitlab* erabili da baina *Sourcetree* erreminta gehigarriarekin. Honen azalpen sakonagoa aurrerago egingo da.
5. Kode unitate hau dispositibo jakin bati esleitu zaio software bideratzailearen bitartez.
6. Azkenik, automatizazioa laborategian ezarri da proiektua bukatutzat emanez.

## 4.3. Aplikazioa

Aipatutako kode unitateen emaitza laborategiko operarioarentzako aplikazio bat da. Honetan, interfaze grafiko baten bitartez erabiltzaileak doitasun eta deskarga partzialen entseguak burutu eta neurketak gordetzen ditu. Aurretik aipatu bezala, hau Pythoneko proiektu baten bitartez garatu da *PyCharm* IDE-a erabiliz. Bere egitura Irudia 12-an adierazten da:

Proiektuak lau karpeta nagusi ditu, bakoitza bere eginkizunarekin:





**Irudia 12:** *autolabo\_sensores* proiektuaren egitura IDE-aren barnean

- **docs:** proiektua ulertzeko beharrezkoak dituen dokumentu guztiak gordetzen dituen karpeta. Bertan daude tentsio igogailuari eta doitasun neurgailuari dagozkien dokumentu guztiak, Artecheren programaziorako liburu zuria eta erabiltzailearentzako bi gidaliburu. Lehenengoan Raspberry PI-a martxan nola jarri adierazten da eta bigarreanean aplikazioaren funtzionamendua azaltzen zaio operarioari.
- **src:** karpeta honetan daude programatutako kode fitxategi guztiak eta interfaze grafikoari dagokion fitxategia. Hauen azalpena banan-banan hurrengo ataletan egingo da.
- **tests:** programaren testea burutzeko beharrezko gidaliburua duen karpeta. Bertan gordetzen dira testen emaitzak ere.
- **venv:** hiru fitxategiz osatua.
  - *.gitignore*: testu fitxategi bat Git bertsio kontrolagailuari adierazten diona zein fitxategi edo karpeta alde batera utzi behar dituen. Defektuz sortzen da.
  - *README.md* : *.md* motako fitxategia. Bertan proiektu honetan lan egiten hasteko beharrezko informazio guztia biltzen da, baita programaren instalaziorako beharrezko baldintzak ere. Bere funtzio nagusia beste programadore batek kode unitateak aldatu edo zerbait gehitu nahi badu proiektuaren lehen ideia bat ematea da.
  - *requirements.txt* : *.txt* motako fitxategia. Proiektuan instalatutako kanpo dependentzia guztiak biltzen dituen fitxategia. Honi esker proiektua instalatzean beharrezkoak diren liburutegiak automatikoki instalatzen dira.



### 4.3.1. Komunikazio moduluak

Komunikazio moduluak informazioa bidali eta biltzeko sortutako klaseak dira. Hauen barnean bi bereizmen nabarmen egin beharra dago, ekipoekin komunikatzeko edo datu baseekin komunikatzeko diren arabera. Ekipoekin komunikatzeko sortutakoak bi dira: *zera.py* eta *hipotronics.py*. Datu baseekin komunikatzeko aldiz bakarra dago: *comunicacion\_web\_service.py*.

#### PuenteZera Klasearen definizioa

*ZERA WM3000U* dispositiboaren erabilera eta kontrolerako sortutako klasea da. Bertan funtzio ezberdinak biltzen dira dispositibora konektatu eta deskonektatzeko, konfigurazioa bidaltzeko eta datuak irakurtzeko.

Konexioari dagokionez TCP/IP erako konexioa ezartzen da [7]. Horretarako nahikoa da Ethernet kable baten bitartez Raspberry PI-a eta Zera konektatzea eta pythoneko *socket* liburutegia [9] erabiltzea. Beraz, konexio hau lortzeko eta konfigurazioari eta irakurketei dagozkien datuak gordetzeko PuenteZera objektu motako klasea sortu da atributu ezberdinekin A Eranskinean 10-99 lerroen artean ikusi daitekeen moduan.

Behin klasea sortuta eta ondo definituta dagoela konexioa ezartzeko eta deskonektatzeko *conectar\_zera()* eta *desconectar\_zera()* funtzioak definitu dira. A Eranskina 100-114 lerroak.

- PuenteZera.**conectar\_zera()**: ZERA dispositiboarekin TCP/IP motako konexioa ezartzen duen funtzioa *socket* objektu baten bitartez, PuenteZera.s. Konexioa zuzena bada PuenteZera.**connected** atributuari 1 balioa ezartzen dio, bestela *socket.timeout* salbuespena sortzen du.
- PuenteZera.**desconectar\_zera**: PuenteZera.s *socket*-a ixten duen funtzioa. Horrela, ZERA dispositiboarekin dagoen konexioa eteten da. PuenteZera.**connected** atributuari 0 balioa ezartzen dio.

Bukatzeko dispositiboari komando ezberdinak bidaltzeko funtzioak definitu dira klasearen barnean. A Eranskinean irakur daitekeen klasearen dokumentazioan azaltzen den bezala komando hauek *SCPI* [19] formatua jarraitzen dute. Entseguei dagokienez beharrezkoa da frekuentzia eta tentsio primario eta sekundarioak konfiguratzea. Hiru komandoek egitura berdina jarraitzen dute eta bakoitzarentzako funtzio bat definitu da. Adibidez frekuentziari dagokiona A Eranskinean 189-221 lerroen artean ikus daiteke.

- PuenteZera.**configurar\_frecuencia**: PuenteZera.**frecuencia** atributuan gordetako datua bidaltzen duen funtzioa PuenteZera.s.**send(AGINDUA)** *socket* objektuaren funtzioaren bitartez. AGINDUA SCPI formatuan kodifikatutako datua izan behar da.

Erroreren bat gertatuz gero *socket.timeout* erako salbuespena sortzen du.

Irakurketari dagokion funtzioak formatu berdina jarraitzen du, A Eranskina 280-298 lerroak, baina kasu honetan beharrezkoa da konexioaren bitartez dispositiboak erantzuten diguna gordetzea, hau izango baita doitasun entseguko neurketaren emaitza.

- PuenteZera.**leer**: neurketaren datuak eskatzen dituen funtzioa. Hauek PuenteZera.s.**recv(BUFFERSIZE)** funtzioaren bitartez eskuratzen dira eta PuenteZera.**data** atributuan gordetzen dira.

Erroreren bat gertatuz gero *socket.timeout* erako salbuespena sortzen du.

### Hipotronics Klasearen definizioa

*HIPOTRONICS 700-DI* dispositiboaren erabilera eta kontrolerako sortutako klasea. Klasearen definizioa eta konexioa ezartzeko funtzioak B Eranskinean 16-126 lerroen artean adierazitakoak dira. *PuenteZera* klasearenekin konparatuz gero aldatzen den bakarra bidaltzen zaizkion komandoen formatua da. Komando hauen bitartez bit kopuru zehatzeko datuak bidaltzen zaizkio dispositiboari orden jakin batean [11], ordena ondokoa da:

START BYTE -> LENGTH -> FUNCTION CODE -> DATA -> CHECKSUM -> END BYTE

- *Start byte* eta *End byte* berdinak dira komando guztientzat:
  - *Start byte*: 0x3A
  - *End byte*: 0x00
- *Length*, *Data*-ren luzera izango da, komandoan ez bada daturik bidaltzen *Length* 0 izango da.
- Komando bakoitzak bere *Function code* propioa izango du.
- *Data*: beharrezkoa den kasuetan bidaltzen diren datuak.
- *Checksum*: aurretik dauden datu guztien biderketa hexadeximala.

Adibidez, maldaren datuak bidaltzeko komandoaren azalpena Taula 4-n ikus daiteke eta deskribapen hau erabiliz *set\_steps()* funtzioa definitu da B Eranskinean 129-165 lerroetan ageri den bezala.

- *Hipotronics.set\_steps*: tentsio maldaren datuak bidaltzeko funtzioa. *Hipotronics.rampN*, *Hipotronics.dwellN* eta *Hipotronics.targetVoltageN* atributuetan gordetako balioak bidaltzen ditu *Hipotronics.s.send(BYTE)* aginduaren bitartez.

Erroreren bat gertatuz gero *socket.timeout* erako salbuespena sortzen du.

Formatu berdina jarraituz funtzio ugari definitu dira klasearen barruan, guztiak B Eranskinean eskuragarri, garrantzitsuenak hauek izanik:

- *Hipotronics.set\_test\_mode\_auto()*: dispositiboa modu automatikoan konfiguratzeko funtzioa.
- *Hipotronics.set\_trip\_voltage()*: *Hipotronics.tripVoltage* atributuan gordetako datua bidaltzeko funtzioa.
- *Hipotronics.set\_trip\_current()*: *Hipotronics.tripCurrent* atributuan gordetako datua bidaltzeko funtzioa.
- *Hipotronics.send\_measurement\_data()*: dispositiboa neurtzen dagoen tentsioa eta intentsitatea eskuratzeko funtzioa.
- *Hipotronics.set\_hv\_on()*: tentsio igotzen hasteko agindua bidaltzen duen funtzioa.
- *Hipotronics.set\_hv\_off()*: tentsio igoera gelditzeko agindua bidaltzen duen funtzioa.

| CMD_SET_STEPS: Request Packet |                   |         |                                |
|-------------------------------|-------------------|---------|--------------------------------|
| Start Byte                    | 0x3A              | uint8_t |                                |
| Length                        | N                 | uint8_t |                                |
| Function Code                 | 0x10              | uint8_t |                                |
| Data                          | N Bytes           | ASCII   |                                |
|                               | Ramp 1            | ASCII   | Format (kV/s): "5.5" = 3 Bytes |
|                               | *                 | ASCII   | Value Separator                |
|                               | Target Voltage 1  | ASCII   | Format (kV): "5.5" = 3 Bytes   |
|                               | *                 | ASCII   | Value Separator                |
|                               | Dwell 1 (seconds) | ASCII   | Format (kV): "5.5" = 3 Bytes   |
|                               | &                 |         | Step Separator                 |
|                               | ...               |         | Up to 15 Steps                 |
|                               | Ramp n            | ASCII   | Format (kV/s): "5.5" = 3 Bytes |
|                               | *                 | ASCII   | Value Separator                |
|                               | Target Voltage n  | ASCII   | Format (kV): "5.5" = 3 Bytes   |
|                               | *                 | ASCII   | Value Separator                |
|                               | Dwell n (seconds) | ASCII   | Format (kV): "5.5" = 3 Bytes   |
| Checksum                      | 0xXX              | uint8_t | As Calculated                  |
| End Byte                      | 0x0D              | uint8_t |                                |

Taula 4: CMD\_SET\_STEPS komandoaren deskribapena [11]

### ComunicacionWebService klasearen definizioa

Web service bitartez Artecheko *DIS3* eta *SAIO* datu baseekin komunikatzeko sortutako klasea. Klaseak balizko erabiltzaile eta pasahitz baten bitartez sarbidea ematen du <https://saio.jakiten.com/Ensayos/WebServices/wsPuntosPrecision.asmx?wsdl> web service-era eta datuak jaso eta bidaltzea ahalbidetzen du. Datuak xml formatuan jasotzen dira eta klasearen barnean funtzio ezberdinak definitu dira xml hauek jasotzeko eta manipulatzeko.

Beraz, esan genezake klasearen barnean bi funtzio mota definitu direla: komunikaziorako funtzioak eta datuak manipulatzeko funtzioak. Dena dela, lehenengo pausua beti da web service-a irekitzea eta horretarako definitutako funtzioa **C** Eranskinean 320-355 lerroen artean agertzen da.

- **ComunicacionWebService.abrir\_ws(usuario,contrasena)**: *Client* motako objektu baten bitartez web servicearekin konexioa ezartzen duen funtzioa. Konexioa ezartzerakoan **ComunicacionWebService.client.service.ComprobarAutenticacion** funtzioari deitzen dio, web-service-aren funtzio propio bat.

Sarrerako parametroak:

usuario – erabiltzailea  
 contrasena – pasahitza

Irteerako parametroak:

respuesta – erantzun boolearra. **True** erabiltzaile eta pasahitza zuzenak badira eta **False** bestela.

Behin konexio hau eginda datuak eskatzeko eta idazteko gai izango gara *SAIO* datu ba-

sean eta datuak eskatzeko soilik erabiliko da *DIS3* datu basea. Jarraitu baino lehen beharrezkoa da argitzea bi hauen ezberdintasuna. Bai datuak irakurtzeko eta bai entseguko emaitzak gordetzeko funtzioak [C](#) Eranskinean daude eskuragarri.

- *DIS3*: dispositibo mota bakoitzari dagokion informazio guztia gordetzen duen datu basea. Datu hauen barnean daude dispositiboaren informazio orokorra (fabrikazio agindua, bezeroa, materiala...) eta entseguetarako datuak (limiteak, kargen balioak,...). Datu base hau irakurketarako soilik erabiltzen da.
- *SAIO*: entseguetako emaitzak gordetzen dituen datu basea. Bertan irakurtzeaz gain idatzi ere egin da.

Atal honekin bukatzeko, aipatu bezala klasearen barnean xml-ak manipulatzeko funtzioak definitu dira. Hauen kasuan datuak errazago manipulatzeko eta gordetzeko hiztegiak erabiltzea erabaki da. Adibidez, *SAIO* datu basetik datorren xml-a irakurtzeko eta beharrezko datuak klasearen atributuetan gordetzeko funtzioa [C](#) Eranskinean 571-636 lerroen artean adierazten da.

- `ComunicacionWebService.leer_xml_ensayo(ensayo:` Saio datu basetik xml fitxategia irakurtzen duen funtzioa entsegu motaren arabera. Ondoren, hiztegi bilakatzan du (datos) eta dagokion atributoaren barnean gordetzen ditu datuak. Adibidez:

```
ComunicacionWebService.errorModulo1 = datos[0]["ERROR_MODULO"]
```

Sarrerako parametroak: ensayo – entsegu mota

#### 4.3.2. Estatu makina

Artecheko implementazioen barnean oso ohikoa da estatu makina bat aurkitzea. Hori dela eta, proiektu honen programa nagusiaren kontrola burutzeko estatu makinaren teoria jarraitzen duen klase bat sortu da *MaquinaEstados* deitua eta *estados.py* fitxategian bildua. Klase honen barruan aurki ditzakegu bere estatu ezberdinak, bakoitzean exekututzen diren ekintzak etab.

Beraz, *estados.py* fitxategiaren barnean ondokoa aurki dezakegu:

- *Enum* motako klase habiaratu baten definizioa. Klase honetan definituta egongo dira estatu makinak har ditzakeen egoerak. Oso erabilgarria da egoeren definizioa leku bakar batean izatea, horrek begirada batean haren funtzionamendua ulertzeko aukera ematen baitu. Kasu honetan hamar egoera ezberdin definitu dira [D](#) Eranskinean 11-26 lerroen artean adierazten den bezala.
- Eraikitzailea, non, batez ere, klasean erabili beharreko aldagaiak hasieratzen diren. Horren barruan sartzen da estatu makinak hartu behar duen hasierako egoeraren definizioa. [D](#) Eranskina 28-57.
- Egoeren arteko trantsizioa egiteko funtzioa. Funtzio honek trantsizioak argi adierazten ditu, eta, gainera, logging erregistro bat uzten da trantsizioa gertatzen den bakoitzean. [D](#) Eranskina 59-76.
- *run()* izeneko metodo bat, egoera-makinaren «tick» -a irudikatuko duena. Hau da, egoera-makinak lana egin dezan nahi denean deituko zaio metodo horri. Metodo honen egitura oso monotona da eta egiten duen bakarra gauden egoera konprobatu

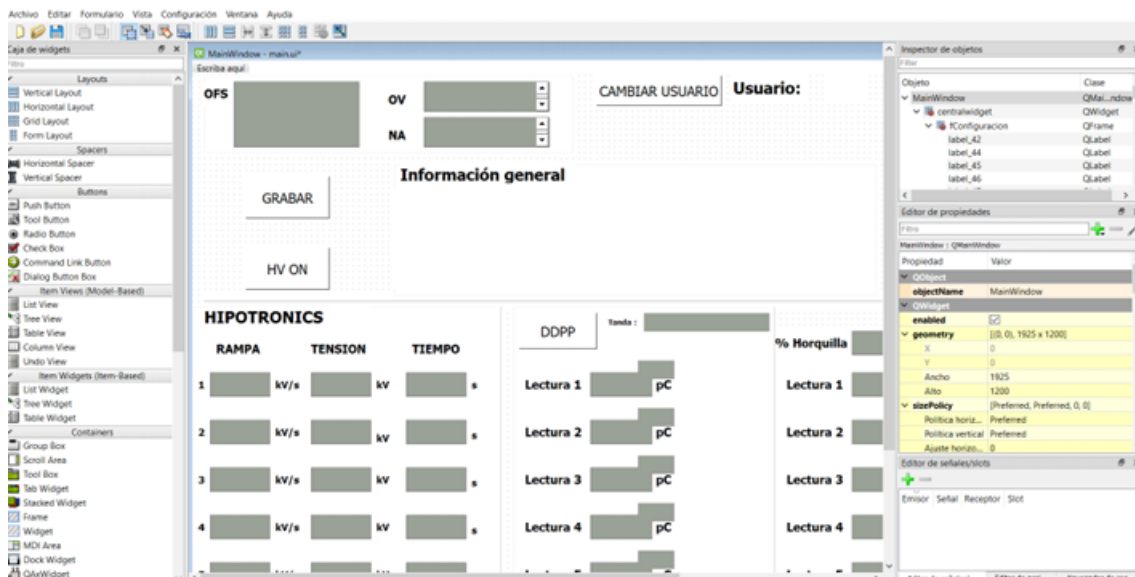
eta egoera horri dagozkion funtzioak exekutatzeko da D Eranskinean 78-131 lerroen artean ageri den bezala.

Programa nagusian E Eranskinean 181-188 lerroen artean ageri den bezala hasieratu eta exekutatu da estatu makina. Exekuziorako *PyQt* liburutegiaren *Timer* [12] bat erabili da. Honi esker *run()* funtzioa exekutatzen da timerraren ziklo bakoitzean. Kasu honetan ziklo hori 10ms-ro burutzeko konfiguratu da.

### 4.3.3. Interfaze grafikoa

Esan bezala, interfaze grafikoa garatzeko *PyQt* liburutegia erabili da eta hori osatzeko *Qt Designer* erremintaz baliatu da. Interfaze grafikoak leiho nagusi bat eta bigarren mailako leihoak ditu. Leiho nagusia *Qt Designer* erabiliz sortu da eta *PyQt* liburutegiarekin konfiguratu da kodean. Bigarren mailako leihoak aldiz, liburutegiaren bitartez kodean bertan bai sortu eta bai konfiguratu dira.

Beraz, alde batetik, *main.ui* fitxategia sortu da non programaren leiho nagusia garatu den *Qt Designer* bitartez. Erreminta hau oso sinplea da eta objektuei zuzendutako programazioan du oinarria. Bertan objektu mota ugari daude eta leiho baten gainera horiek arrastatuz nahi den diseinua sor daiteke. Irudia 13-ri erreparatuz gero, ikus daiteke ezkerreko zutabearen objektu moten zerrenda ageri dela. Bada bertatik nahi dena aukeratzeko da eta erdiko pantaila nagusira eramaten da. Behin hori eginda eskumako zutabearen agertzen da eta bertan izendatu eta konfiguratzeko aukerak ematen dira. Izendapenak oso garrantzitsuak dira gero programa nagusian erabiliko direlako. Horrez gain, objektua aukeratzeko orduan kontuan eduki behar da bere funtzionalitatea zein izango den, objektu bakoitzak definitutako metodoak baititu *PyQt* liburutegian.



Irudia 13: *main.ui* fitxategia *Qt Designer* erremintan

Erabilitako objektu garrantzitsuenak ondokoak izan dira:

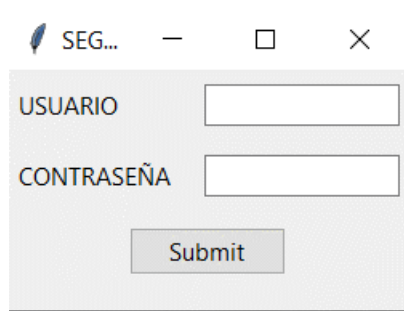
- **QLabel**: irudiak edo testuak erakusteko erabilia. Ez du erabiltzailearekin iteraziorik ahalbidetzen. Deskribapen sakonagoa eta funtzio erabilgarriak bere dokumentazioan aurki daitezke [13].

- **QTextEdit**: irudiak, zerrendak eta taulak erakutsi ditzakeen objektua. Optimizata dago dokumentu handiak maneiatzeko eta azkar erantzuteko erabiltzailearen sarrerei. Deskribapen sakonagoa eta funtzio erabilgarriak bere dokumentazioan aurki daitezke [14].
- **QPushButton**: akzio agindu bat bidaltzeko edo erantzun bat lortzeko erabiltzen den botoia. Honek, seinale bat emititzen du aktibatzen denean. Aktibatzeko arratoiarekin klikatu daiteke edo teklatuko botoi bat konektatu ahal zaio [15]; horrela, teklatuko botoi hori sakatzean interfazeko botoia klikatzearen berdina izango da. Oso garrantzitsua da botoiari guk nahi diogun funtzioa konektatzea edo esleitzea. Hau programa nagusian egin da `configurar_botones()` funtzioa erabiliz. E Eranskina 1331-1361.

Kode lerro horietan fijasen bagara botoien estilo konfigurazioa kodean bertan egin da. Izan ere, estilo konfigurazio guztia kodearen bitartez egin da bai koloreak eta bai letra-tipoa korporatiboak izan daitezzen. Gainera, estatu makinaren egoeraren arabera interfazea aldatuz doa botoiak aktibatuz eta desaktibatuz eta koloreak aldatuz. Guzti hau programa nagusian egin da E Eranskinean 31-234 lerroen artean.

Beste alde batetik, soilik kode bitartez sortutako bigarren-mailako leihoak ditugu bakoitza bere eginkizun eta itxura konkretuarekin:

1. **Log in leihoa**. Erabiltzailea eta pasahitza eskatzeko sortutako leihoa. `VentanaLogIn` deituriko klase baten bitartez definitu da (F Eranskina) eta bere itxura Irudia 14-n ikus daiteke.

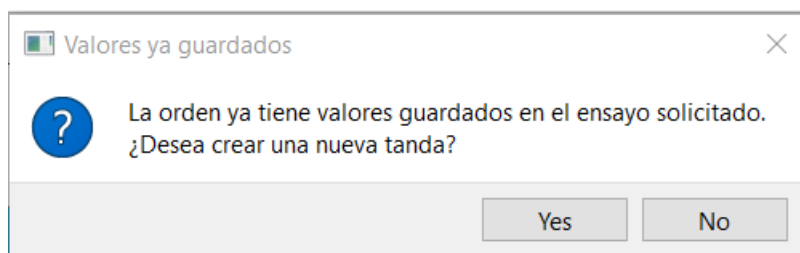


**Irudia 14:** *Log in leihoa*

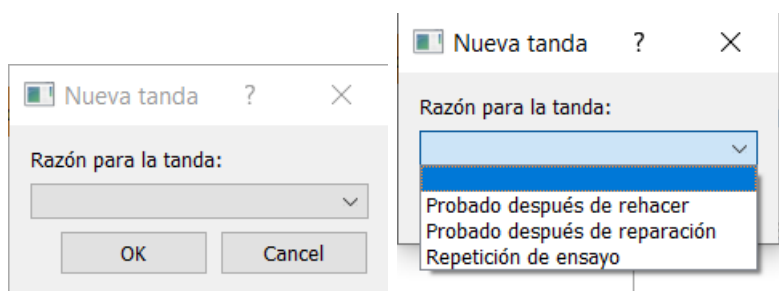
Erabiltzaileak erabiltzailea eta pasahitza idatzi behar ditu eta ondoren “Submit” botoia sakatu.

2. **Galdera leihoa**. Entsegu txanda berri bat egin nahi den galdetzeko sortutako leihoa. Hau programa nagusian definitu da `QMessageBox` [16] erako `Qt` objektua erabiliz eta bere itxura Irudia 15-en ikus daiteke. Erabiltzaileak “Yes” edo “No” aukeratu behar du. Bere definizioa kodean E Eranskinean 2833-2856 lerroen artean ikus daiteke.
3. **Aukera leihoa**. Entsegu txanda berria egiteko arrazoen aukerak biltzen dituen leihoa. Hau programa nagusian definitu da `QInputDialog` [17] erako `Qt` objektua erabiliz eta bere itxura Irudia 16-en ikus daiteke ematen diren aukerekin batera. Bere definizioa kodean E Eranskinean 2858-2878 lerroen artean ikus daiteke.

Erabiltzaileak hiru aukeretako bat aukeratu behar du derrigorrez eta ondoren “Ok” botoia klikatu. “Cancel” botoia sakatuz gero galdera leihoa agertzen da berriz ere.

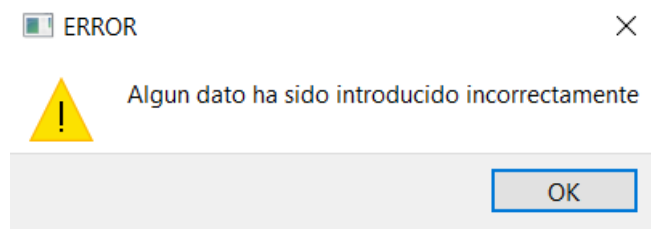


**Irudia 15:** *entsegu txanda berri bat nahi den galdetzeko sortutako leihoa*



**Irudia 16:** *txanda berrirako aukera leihoa*

4. **Errore leihoak.** Edozein errore gertatzen bada agertzen diren leihoak. Hauek programa nagusian definitu dira *QMessageBox* [16] erako *Qt* objektua erabiliz. Errore mota bakoitzarentzako izenburu eta testu ezberdin bat adierazi da. Adibidez erabiltzailea edo pasahitza gaizki sartuz gero erakusten den mezua Irudia 17-en adierazten da. Leiho orokorraren definizioa kodean E Eranskinean 2800-2817 lerroen artean ikus daiteke.



**Irudia 17:** *log in okerraren errore leihoa*

#### 4.3.4. Programa nagusia

Azkenik programa nagusia daukagu, aurreko klase eta interfaze guztiak biltzen dituen *App* klasearen barnean *main.py* fitxategian. Bere funtzionamendu orokorra G Eranskineko Fluxu-diagrama 1-en ageri da eta klase osoaren definizioa E Eranskinean biltzen da. Bertan azpi-prozesu ugari bereiz daitezke eta bakoitzak aldaketak eragiten ditu interfaze nagusian edo bigarren-mailako leihoak azalarazten ditu.

Behin aplikazioa zabaltzen dugunean emaria ondokoa izango da:

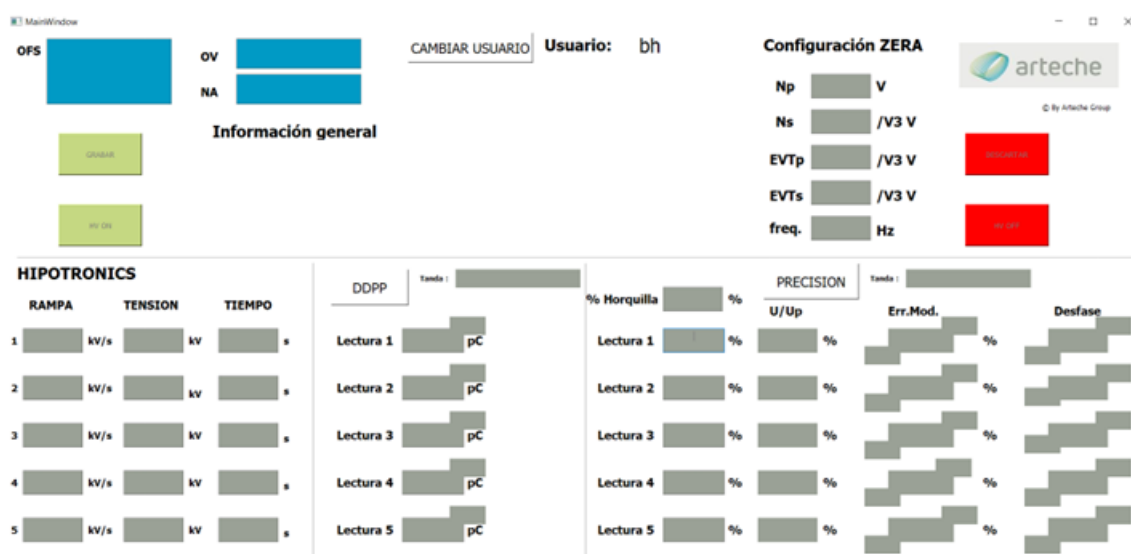
1. **Log in:** erabiltzailea eta pasahitza eskatzeko azpi-prozesua. Bertan jarraitzen den logika G Eranskineko Fluxu-diagrama 2-n ageri da.

Azpi-prozesu honetan *Log in* leihoa azalarazten da eta behin web-servicerako goiburua sortuta autentifikazioa egiaztatzen da; hau da, ea erabiltzaile eta pasahitza



zuzenak diren. Hau horrela izan ezean errore leiho bat agertzen da eta aplikazioa *Log in* leihoan uzten du. Aldiz, zuzenak badira interfaze nagusia bistaritzen da erabiltzailea adieraziz **Usuario** eremuan. Horrez gain **OFs** eremua aktibatzen da, erabiltzaileari bertan idazten utziz eta **DDPP**, **PRECISION** eta **CAMBIAR USUARIO** botoiak aktibatzen dira Irudia 18-n ageri den bezala:

- **OFs**: fabrikazio ordena (OF) idazteko eremua. Fabrikazio ordena 11 digituz osaturiko zenbakia da. Fabrikatuko den sentsore bakoitzak berea dauka eta bakarra da.
- **DDPP**: deskarga partzialen entsegua adierazten duen botoia.
- **PRECISION**: doitasun entsegua adierazten duen botoia.
- **CAMBIAR USUARIO**: erabiltzailea aldatzeko botoia.



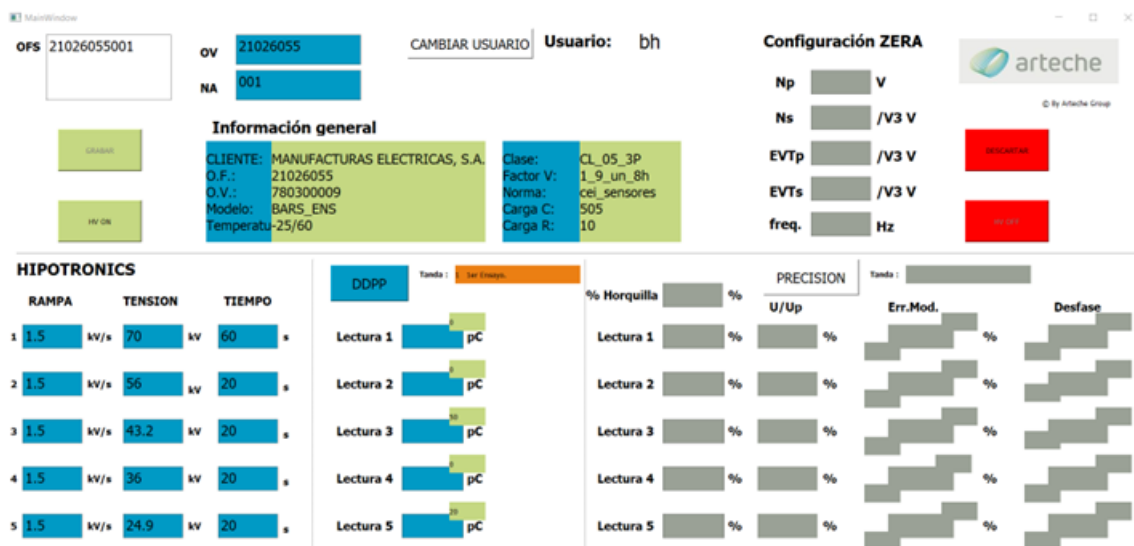
Irudia 18: leiho nagusia *Log in* prozesua bukatzean

2. **Entsegu mota aukeratu**: erabiltzaileak *OF*-a idatzi eta entsegu mota aukeratzeko du. Aukeraketa hori egiteko **DDPP** edo **PRECISION** botoian klikatu behar da. Botoia klikatzen den momentuan **G** Eranskineko Fluxu-diagrama 3-n ageri den prozesua martxan jartzen da.

Lehenik *OF*-a zuzena den egiaztatzen du. Ondoren, entseguko datuak irakurtzen ditu **G** Eranskineko Fluxu-diagrama 4-ko azpi-prozesua jarraituz. Bertan adierazten da irakurketa egokia ez bada errore pantaila agertzen dela adieraziz. Hau egin ostean, entseguak jada balioak gordeta dituen konprobatzen da eta horrela izanez gero galdera leihoa agertzen da. Aurretik azaldu den bezala galderari baietz erantzuten bazaio aukera leihoa agertzen da. Prozesu guztia amaitzean pantaila nagusira itzultzen gara baina pantaila eguneratuarekin.

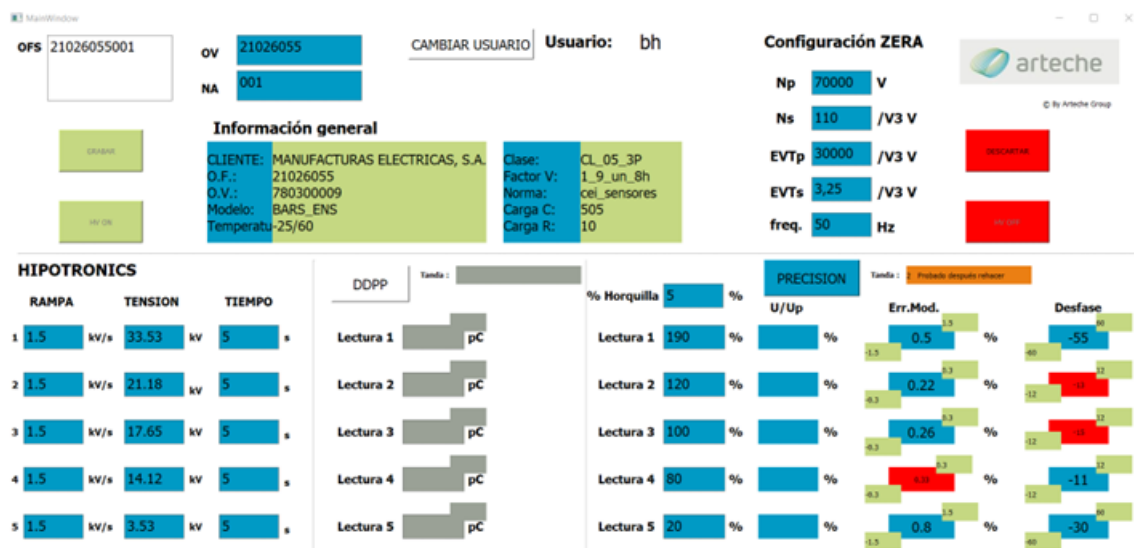
Adibidez, Irudia 19-ri erreparatzen badiogu *21026055001 OF*-aren deskarga partzialen entsegua burutzea aukeratu da. Kasu honetan lehenengo entsegua da Tanda eremuan ikusi ahal dugun bezala eta ez dago baliorik gordeta. **HIPOTRONICS** eta **INFORMACIÓN GENERAL** eremuei dagozkien informazioa bete da lortutako datuak erabiliz, baita entseguren limiteei dagokiena ere. Horrez gain **HV ON** eta **DESCARTAR** botoiak aktibatu dira.





Irudia 19: 21026055001 OF-a idatzi eta DDPP botoia sakatzean programa nagusiaren itxura.

Aldiz, Irudia 20-en OF berdinentzako doitasun entsegua burutzea aukeratu da. Kasu honetan 2.txandan gaude eta balioak dauzka jada gordeta. Beraz entsegu txanda berria egin nahi dugun galderari baietz erantzun zaio, arrazoia “*Probado después de rehacer*” izanik. Gainera, txanda berri hau burutu da eta balioak gordeta ditu beraz berriz agertu da galdera leioa eta oraingoan ezetz adierazi diogu. Ohartzen bagara, **HIPOTRONICS** eremuko balioak ezberdinak dira eta **Configuración ZERA** eremua aktibatu eta bete da, doitasun entseguaren datuen eremuez gain. Gainera, berriz ez entseatzea erabaki denez **DESCARTAR** botoia soilik dago aktibatuta. Bukatzeko, limiteetatik kanpo dauden balioak gorritz adierazten dira.



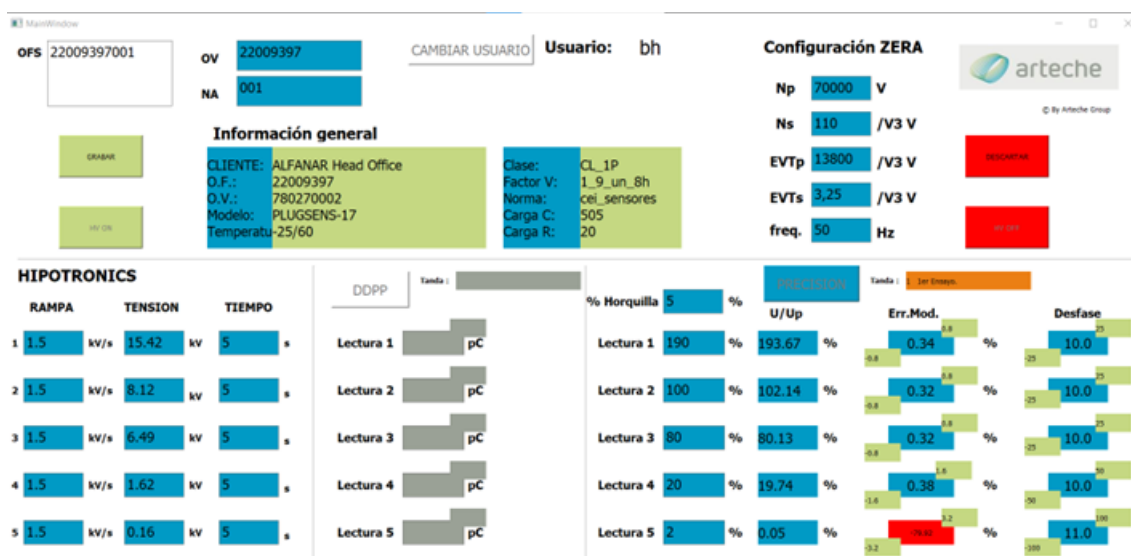
Irudia 20: 21026055001 OF-a idatzi eta PRECISION botoia sakatzean programa nagusiaren itxura.

3. Entsegua hasi: HV ON botoia sakatzean hasten den prozesua, G Eranskineko Fluxu-diagrama 5-en azaltzen da.

Lehen pausua tentsio igogailuarekin konexioa ezartzea da. Konektatuta dagoenean modu automatikoan jartzeko agindua bidaltzen zaio konfigurazioarekin eta tentsioa igotzen hasteko aginduarekin batera. Konexioa burutzen ez bada errore pantaila agertzen da baina entsegua jarraitzeko aukera dago. Horretarako, erabiltzaileak eskuz egin beharko ditu aurreko pausuk.

Bigarren pausua zein entsegu mota aukeratu den konprobatzea da. Izan ere, deskarga partzialen kasuan neurgailua ez dagoenez automatizatuta langileak eskuz sartu behar ditu datuak. Doitasunaren kasuan aldiz, doitasun neurgailuarekin konexioa ezartzea da hurrengo pausua. Hau burutu ezean entsegua bukatutzat ematen da.

Konexioa zuzena bada konfigurazioa bidaltzen zaio eta neurketak irakurtzeko eta idazteko prozesua hasten da. Hau da, tentsioa neurketa puntuetako batean egonez gero, modulu zein angelu erroreak pantailan idazten dira neurtu diren tentsioaren balio zehatzarekin batera. Irudia 21. Entsegua bukatuko da puntu guztiak idazten direnean edo **HV OFF** botoia sakatzen denean. Bi kasuetan **GRABAR** eta **DESCARTAR** botoiak aktibatzen dira eta hurrengo prozesura igarotzen da programa.

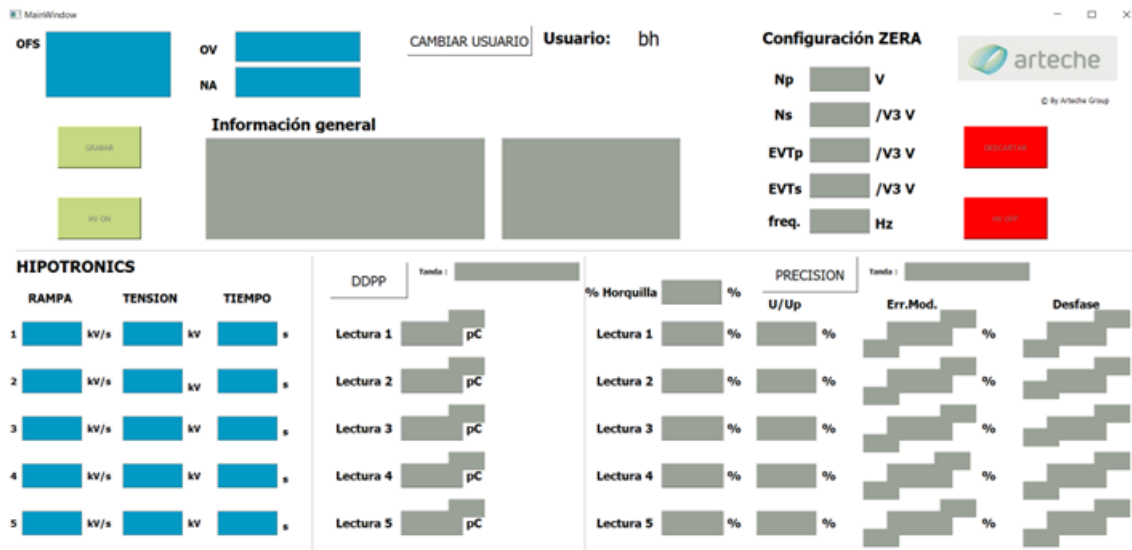


Irudia 21: doitasun entsegua bukatzean pantaila nagusiaren itsura

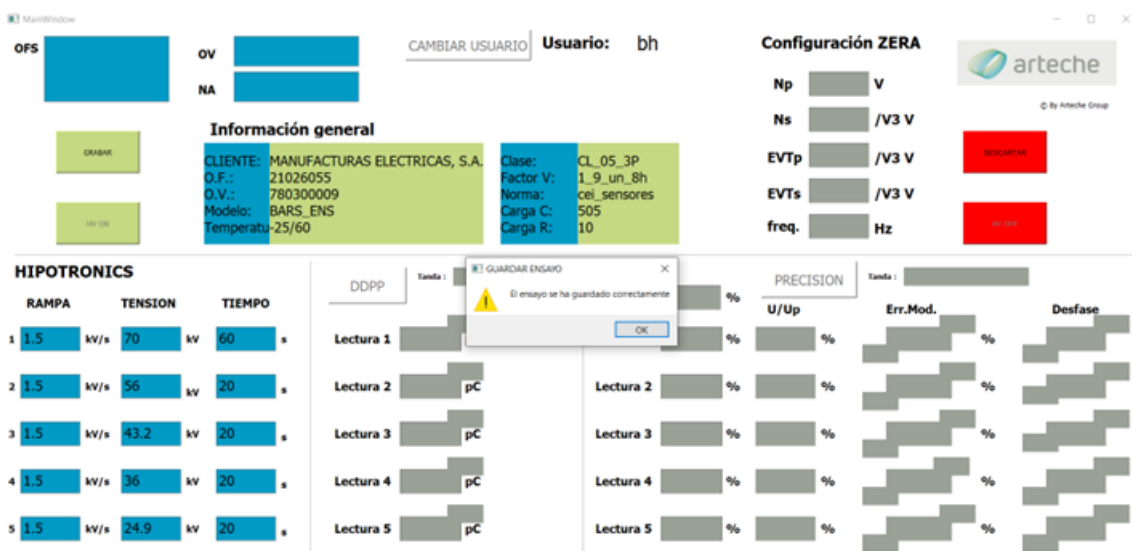
4. **Entseguaren bukaera:** programa nagusiaren azken atala non dispositibo guztiak deskonektatzen diren **G** Eranskineko Fluxu-diagrama 6-en ageri den sekuentzia jarraituz.

Bertan bi aukera daude: entseguaren emaitzak datu basean gordetzea edo datuak baztertzea. Alde batetik, baztertzeko **DESCARTAR** botoia klikatu behar da eta pantaila guztia garbitzen da Irudia 22-n ageri den bezala. Bestetik, **GRABAR** botoia klikatuz gero datuak **SAIO** datu basean gordetzen dira eta ondo gorde direla adierazteko pantaila agertzen da Irudia 23-n bezala.

Puntu honetan prozesua bukatutzat eman dezakegu eta programa entsegu motaren aukeraketa puntura itzultzen da non **OF** berri bat idatzi behar duen **G** Eranskineko Fluxu-diagrama 1-en ageri den bezala. Aipatu beharrekoa da **CAMBIAR USUARIO** botoia sakatzen bada *Log in* puntura itzultzen dela programa.



Irudia 22: leiho nagusia DESCARTAR botoia klikatzean



Irudia 23: leiho nagusia GRABAR botoia klikatzean eta datuak ondo gorde direnean

## 4.4. Bertsioen kontrola

Bertsioen kontrola, iturburu-kontrola izenez ere ezagutzen dena, software-kodean aldaketak bilatu eta administratzeko erabiltzen da. Bertsioak kontrolatzeko sistemak software-tresnak dira, eta software-ekipoei laguntzen diete iturburu-kodean denboran zehar aldaketak kudeatzeko. Garapen-inguruneak azkartu ahala, bertsioak kontrolatzeko sistemek software-ekipoei laguntzen diete azkarrago eta adimentsuago lan egiten.

Bertsioak kontrolatzeko softwareak kodearen aldaketa bakoitzaren jarraipena egiten du datu-base mota berezi batean. Akats bat eginez gero, garatzaileek atzera itzul dezakete erlojua, eta kodearen aurreko bertsioak konparatu, akatsa konpontzen laguntzeko, eta, aldi berean, taldeko kide guztiei ahalik eta eten gutxien egiteko. Bertsioen kontrola, beraz, software-kodean aldaketak bilatu eta kudeatzea da [18].

Hau esanda, proiektu honen bertsio kontrola burutzeko **Git** erabili da **Sourcetree** erremintaz baliatuz. Kasu honetan, programatzaile bakarra egon denez honen helburu nagusia aldaketak ondo dokumentatuta uztea izan da, eraman den prozesuarekin batera. Beste helburuetako bat Artecheko *Sistema Informatikoen Departamentuak* kodea eskuragarri edukitzea izan da, beharrez gero begirada bat botu eta aldaketak eskatzeko edo kodea beste proiektu batzuetan erabiltzeko aukera izateko.

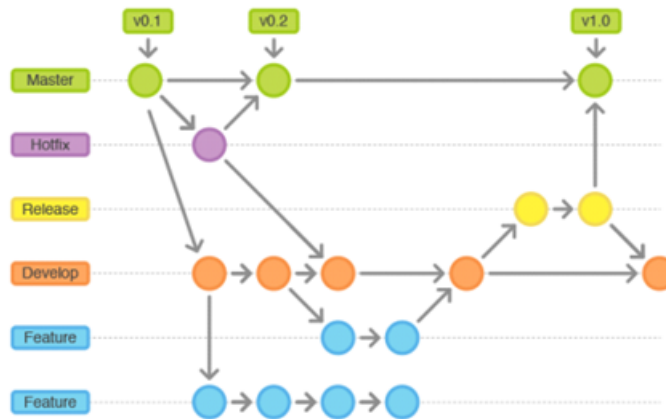
Dokumentazioa burutzeko *AIDA* proiektuaren barruan zehaztutako arauak jarraitu dira:

- Aldaketa txiki bat egin bezain pronto “*commit*” bat burutu behar da dagokion dokumentazioarekin.
- “*commit*” deritzonaren estruktura ondokoa izango da:
 

```
<mot> [esparrua (aukerazkoa)]: azalpen laburra
[azalpen sakona (aukerazkoa)]
[oina (aukerazkoa)]
```
- Motak ondokoak dira:
  - **Feat**: ezaugarri berri bat sartzen du. *GitFlow*-eko *feature* adarrarekin korrelazioan dago.
  - **Fix**: akats bat zuzentzen du.
  - **Style**: estiloarekin lotutako aldaketak. Formatu aldaketak, tabulazioak, espazioak... , funtzionamenduari eragiten ez diotenak.
  - **Refactor**: kodearen funtzionaltasuna aldatzen ez duten aldagaien edo funtzioen izen-aldaketak.
  - **Test**: test bat gehitu da.
  - **Docs**: aldaketak dokumentazioan.
  - **Perf**: errendimendua hobetzeko aldaketak burutu dira.
- Adar berriak sortzen joatea komeni da egin nahi denaren arabera. Adibidez, laborategian implementatuta dagoen bertsioa *Master* edo **Main** adarrean egongo da. Aldiz, aldaketak **Develop** adarrean egiten dira eta bertatik adar berriak sortu behar dira egiten gabiltzanaren arabera. Funtzionalitatea gehitzen bagabiltza **Feature**

adar berri batean egin behar da eta aldiz produkzioko errore bat zuzentzeko **Hotfix** adarrean zuzendu behar da. **Release** adarra testak egiteko da.

Adarren erabilera hobeto ulertzeko proiektu ez erreal baten errepresentazio bat Irudia 24-en ageri da:



**Irudia 24:** *GitFlow-eko adarren errepresentazio adibidea [20]*

Azkenik, *Sourcetree* aplikazioan proiektu jakin honen fluxuaren zati bat Irudia 25-en ageri da. Bertan bereiz daiteke **Main** adarra *v1.0.0* bertsioan dagoena kolore grisez adierazia. Bestalde, **Develop** adarra daukagu urdinez irudikatua eta gorritz *Logger* deituriko **Feature** adarra. Bakoitzak bere “commit” propioak ditu batzuk **Docs** motakoak izanik eta beste batzuk aldiz **Feat** motakoak.



**Irudia 25:** *proiektuaren emariaren zati bat, Sourcetree aplikazioan.*

## 5. Ondorioak

Proiektuaren ondorio nagusiak tentsio erdiko sentsoreen laborategian antzemandakoak dira. Bertan ikusi da prozesu bat automatizatzeak dakartzan onurak zeintzuk diren bai denborari eta bai fidagarritasunari dagokionez.

Alde batetik, proiektu honi esker errendimendu hobea lortu da laborategian. Izan ere, hasiera batean operarioa entseguak irauten zuen denbora osoa egon behar zen dispositiboetara begira emaitzak apuntatu ahal izateko. Orain aldiz, sentsorea entseatzen dabilen bitartean beste eginkizun batzuk burutzeko erabili dezake denbora hori: hurrengo sentsorea prestatu, jada entseatutako ekipoak enbalatzen joan, postua ordenatu...

Beste alde batetik, erroreak gutxitzea eta emaitzen fidagarritasuna handitzea lortu da. Kontuan hartu, lehen emaitza guztiak eskuz idazten zirela datu baseetan. Horregatik, akatsak agertzen ziren emaitzetan: zenbakiren bat gaizki idatzita, minus ikurren bat faltan edo koma bat desplazatuta. Aplikazioari esker, emaitzen irakurketa zuzena burutzen da dispositiboetatik akats horiek desagerraraziz. Horrez gain, konfigurazioak irakurri eta berriz ere eskuz idatzi behar ziren bai doitasun neurgailuan eta bai tentsio igogailuan. Orain konfigurazio guztia datu baseetik irakurri eta bidaltzen denez errore horiek ezabatu dira.

Aipatzekoak dira ere, enpresa baten barruan programazio erako proiektu bat aurrera eramateak azaleratzen dituen ondorioak:

- Programazio estilo bateratu bat beharrezkoa da. Honi esker enpresako edozein langile izango da gai kodea ulertzeko jakintza minimo batzuk baditu programazioaren inguruan. Honi esker, aplikazioaren mantentze lanak ez dira sortzailearen eskutik soilik gelditzen. Gainera, Artecheko beste programatzaile bat kodea bererabiltzeko gai izango da beste proiektu batzuetan nahi izanez gero, lana aurreztuz.
- Dokumentazio egokia nahitaezkoa da bai beste programatzaileentzat bai langileenentzat. Berriz ere, programatzaileei dagokienez kodearen ulertze zabalagoak onura argiak ditu. Laborategiko langileei dagokienez, esku-liburu argi eta egokien beharrian argia dago gero aplikazioa era zuzen batean erabiltzen dela bermatzeko eta denbora aurrezteko.
- Bertsioen kontrol egokia funtsezkoa da. Azken finean, pertsona ugari proiektuan lanean hasiko badira egindako aldaketen kontrol zuzena eramatea ezinbestekoa da. Horrez gain, aurreko bertsio zuzen bat edukitzeak askatasun handiagoa ematen du programazioaren ikuspuntutik. Izan ere akatsen bat agertzen bada beti dago aukera aurreko bertsiora bueltatzeko.
- Gaur egun zibersegurtasuna gogoan izan behar dugu. Zibererasoak pil-pilean dauden zerbait dira eta ez badira beharrezko neurriak hartzen edozeinek edukiko luke sarbidea enpresa bateko datu guztietara.

Bukatzeko, esan beharra dago proiektu hau ez dela bukatutzat eman oraindik. Martxan jarri dira proiektuaren barnean beste atal batzuk automatizatzeko eginkizunak: kargen kutxa eta etiketen inpresioa automatizatzeko proiektuak eta sentsoreak blokeka entseatzea ahalbidetuko duen proiektua. Hauen oinarria proiektu honetan garatutako programa bera izango da eta helburuak funtzionalitate berriak gehitzea eta programa bera ahal den heinean hobetzen joatea izango da.

## Erreferentziak

- [1] Artech Group. (2021, azaroak 25). *Kits de Sensores de Media Tensión para la automatización de la distribución*. <https://www.artech.com/es/cmisis/document/default/6d3b19c1-ea3e-45ac-b811-aef529bfc9?uuiid=2c8ca2f9-0840-42b2-89bd-976c46c91d0c>
- [2] Cotteleer, M., Sniderman, B. (d.g.). *Forces of change: Industry 4.0*. Deloitte. <https://www2.deloitte.com/content/dam/Deloitte/es/Documents/manufacturing/Deloitte-ES-manufacturing-industria-4.0.pdf>
- [3] Artech Group. (2021, ekainak 8). *Informe annual 2020*. <https://www.artech.com/es/cmisis/document/default/0f4ce794-80cb-4c09-99b0-0443ecbcc82b>
- [4] Herrera Portilla, W. (2017). *Características técnicas de los transformadores para medida en líneas de media tensión según estándares IEC 61869*. CIDET. [https://www.cidet.org.co/sites/default/files/documentos/art\\_6.pdf](https://www.cidet.org.co/sites/default/files/documentos/art_6.pdf)
- [5] International Electrotechnical Commission. (2017, abendua). *INSTRUMENT TRANSFORMERS-Part 11: Additional requirements for low-power passive voltage transformers (IEC 61869)*
- [6] DRILCO. (2020). *E20-0210. HIPOTRONICS*.
- [7] ZERA. (2017, ekainak 13). *Operation Manual: WM300U Voltage Transformer – Measuring Bridge*.
- [8] OMICRON electronics. (2020). *MPD 800 User Manual*.
- [9] Python (d.g.). *socket – Low-level networking interface*. Docs.python.org. <https://docs.python.org/3/library/socket.html>
- [10] ZERA. (2018). *Transformer Measuring device WM300U SCPI Interface Description*.
- [11] HIPOTRONICS. (2019, martxoak 1). *700-DI Communication Protocol Document*.
- [12] Qt. (d.g.). *QTimer - Qt for Python*. Doc.qt.io. <https://doc.qt.io/qtforpython-5/PySide2/QtCore/Qtimer.html>
- [13] Qt. (d.g.). *QLabel - Qt for Python*. Doc.qt.io. <https://doc.qt.io/qtforpython-5/PySide2/QtWidgets/QLabel.html?highlight=qlabel#more>
- [14] Qt. (d.g.). *QTextEdit - Qt for Python*. Doc.qt.io. <https://doc.qt.io/qtforpython-5/PySide2/QtWidgets/QTextEdit.html?highlight=qtextedit#PySide2.QtWidgets.PySide2.QtWidgets.QTextEdit>
- [15] Qt. (d.g.). *QPushButton - Qt for Python*. Doc.qt.io. <https://doc.qt.io/qtforpython-5/PySide2/QtWidgets/QPushButton.html?highlight=qpushbutton#PySide2.QtWidgets.PySide2.QtWidgets.QPushButton>
- [16] Qt. (d.g.). *QMessageBox – Qt for Python*. Doc.qt.io. <https://doc.qt.io/qtforpython-5/PySide2/QtWidgets/QMessageBox.html?highlight=qmessagebox#PySide2.QtWidgets.PySide2.QtWidgets.QMessageBox>

- [17] Qt. (d.g.). *QInputDialog - Qt for Python*. Doc.qt.io. <https://doc.qt.io/qtforpython-5/PySide2/QtWidgets/QInputDialog.html?highlight=qinputdialog#PySide2.QtWidgets.PySide2.QtWidgets.QInputDialog>
- [18] Bitbucket. (d.g.) *What is version control?*. Atlassian.com. <https://www.atlassian.com/git/tutorials/what-is-version-control>
- [19] Keysight. (d.g.) *Introduction to SCPI language*. Keysight.com. [https://rfmw.em.keysight.com/spdhelpfiles/33500/webhelp/US/Content/\\_\\_\\_I\\_SCPI/00%20scpi\\_introduction.htm#:~:text=SCPI%20%28Standard%20Commands%20for%20Programmable%20Instruments%29%20is%20an,a%20common%20node%20or%20root%2C%20thus%20forming%20subsystems](https://rfmw.em.keysight.com/spdhelpfiles/33500/webhelp/US/Content/___I_SCPI/00%20scpi_introduction.htm#:~:text=SCPI%20%28Standard%20Commands%20for%20Programmable%20Instruments%29%20is%20an,a%20common%20node%20or%20root%2C%20thus%20forming%20subsystems).
- [20] Ofili C. (2018, apirilak 9). *Gitflow Workflow, Automated Builds, Integration & Deployment*. Medium.com. <https://medium.com/devsondevs/gitflow-workflow-continuous-integration-continuous-delivery-7f4643abb64f>



## A. PuenteZera Klasea

```

1 import math
2 import socket
3
4 # VARIABLES GLOBALES A futuro deben tomar los valores de un fichero de
   configuraci n.
5 BUFFERSIZE = 1024 # tama o del buffer de datos para el uso del socket
6
7 TCP_IP = "192.168.0.3" # IP
8 TCP_PORT = 6315 # PUERTO
9
10
11 class PuenteZera(object):
12     """Clase para uso y control del dispositivo ZERA WM3000U.
13
14     Se utiliza la libreria socket para establecer conexi on TCP/IP con el
   dispositivo y as mandar ordenes de cambio de configuraci n y de
   lectura de medidas.
15
16     Es necesario poner el PC y el dispositivo en el mismo area de IP con
   solo el ltimo digito cambiado. Los detalles de una conexi n
   optima est n recogidos en el documento 'WM3000U_MAN_EXT_GB_V41' el
   manual de operaciones del ZERA.
17
18     Los comandos que hay que mandarle al dispositivo est n dise ados en
   formato SCPI y est n dise ados siguiendo los est ndares y las
   reglas de sintaxis de este formato. Todos los comandos est n
   recogidos en el documento 'WM3000U_SCPI_EXT_GB_V105'.
19
20     Ejemplo de comando para pedirle al dispositivo la lectura de las
   medidas:
21         SCPI:READ?'/n
22         Python: self.s.send(b"READ?/n")
23
24     Las respuestas del dispositivo son bytes con los datos correspondientes
   que se le han pedido.
25
26     Ejemplo de respuesta de la lectura:
27         Python: self.s.recv(BUFFERSIZE)= b"Frequency;VectorN;VectorX;
   Loadpoint;Loadpoint1;Error1;Error2;Angle;RCF"
28
29     NOTAS:
30         1.-Al final de cada comando siempre hay que escribir el
   caracter correspondiente de final de linea. En este caso /n
31
32         2.-Al final de los comandos de cambiar configuraci n siempre
   hay que mandar el comando OPC?
33
34         3.-Se recomienda el valor 1024 para el buffersize.
35
36     """
37
38 # Inicializar atributos. Conecatr SOCKET
39 def __init__(self):
40     """Funcion por defecto de inicializacion de objeto dentro de la
   clase. Se les asignan valores de inicializaci n a los atributos
   del objeto.

```

```

41
42     Despu s de la inicializaci n de los parametros correspondientes
43     al ensayo de precisi n conecta mediante un socket el Puente
44     Zera con el PC.
45
46     Parametros:
47         self — clase PuenteZera
48
49     Atributos:
50
51         self.frecuencia — Frecuencia de la se al. Valores
52         posibles: 16.66Hz, 50Hz, 60Hz / Tipo: str
53
54         self.EVTprim — Tensi n primaria del equipo bajo
55         prueba / Tipo:str
56
57         self.EVTsec — Tensi n secundaria del equipo bajo
58         prueba / Tipo:str
59
60         self.Nprim — Tensi n primaria del patr n / Tipo:str
61
62         self.Nsec — Tensi n secundaria del patr n / Tipo:str
63
64         self.Punto — Tensi n del punto a medir % / Tipo:float
65
66         self.U — Tensi n del punto actual en V / Tipo:str
67
68         self.Frecuencia2 — Frecuencia de la medida en Hz /Tipo
69         :str
70
71         self.ErrorPunto — Error aceptable del punto a medir %
72         /Tipo:float
73
74         self.ErrorModulo — Error aceptable del modulo /Tipo:
75         str
76
77         self.ErrorAngulo — Error aceptable en angulo /Tipo:
78         str
79
80         self.LPX — Tension actual del equipo bajo medida /Tipo
81         :str
82
83         self.LPN — Tension actual del patron /Tipo:str
84
85         self.RCF — Ratio correction factor /Tipo:str
86
87         self.connected — #0 si el socket est desconectado y
88         1 si est conectado /Tipo:int
89
90     Excepciones:
91
92         Socket TimeOut — Si el socket tarda m s de 3 segundos
93         en conectarse el programa dejara de intentarlo y se
94         pondr el atributo self.connected a 0 indicando
95         que no se ha realizado la conexi n.
96
97     """
98     # Inicializaci n de atributos definidos:
99     self.frecuencia = "0.0"

```

```

85     self.EVTprim = "0.0"
86     self.EVTsec = "0.0"
87     self.Nprim = "0.0"
88     self.Nsec = "0.0"
89     self.Punto = 0.0
90     self.U = "0.0"
91     self.Frecuencia2 = "0.0"
92     self.ErrorPunto = 5.0
93     self.ErrorModulo = "0.0"
94     self.ErrorAngulo = "0.0"
95     self.LPX = "0"
96     self.LPN = "0"
97     self.RCF = ""
98     self.connected = None
99
100    def conectar_zera(self):
101        # Configuraci3n de la conexi3n del socket(Conexi3n TCP/IP)
102        self.s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
103        # Comienzo de conexi3n con Puente ZERA
104        self.s.settimeout(3)
105        try:
106            self.s.connect((TCP_IP, TCP_PORT))
107            self.set_connected(1)
108        except socket.timeout:
109            self.set_connected(0)
110
111    def desconectar_zera(self):
112        if self.connected == 1:
113            self.s.close()
114            self.set_connected(0)
115
116
117    # Enviar al zera ordenes deconfiguraci3n.
118    def configurar_tensiones_n(self):
119        """Funci3n para configurar las tensiones primaria y secundaria del
120            equipo bajo prueba. Este manda las ordenes de configuraci3n al
121            Puente Zera.
122
123            Parametros:
124                self -> clase PuenteZera
125
126            Atributos:
127                orden -> Orden de configuraci3n que se le envia al PUENTE
128                ZERA
129
130            Excepciones:
131                Socket TimeOut - Si la conexi3n tarda m s de 0.2
132                segundos en recibir la informaci3n se pasa a la
133                siguiente
134                orden de configuraci3n.
135
136            """
137        orden = "CONF:RAT:N " + self.Nprim + "V/w3:" + self.Nsec + "V/w3\n"
138        self.s.send(b"CONF:RAT:N?\n")
139        self.s.settimeout(3)
140        try:
141            self.s.recv(BUFSIZE)
142        except socket.timeout:

```

```

138         pass
139     self.s.send(bytes(orden, "ascii"))
140     self.s.send(b"CONF:APPLY\n")
141     self.s.settimeout(3)
142     try:
143         self.s.recv(BUFFERSIZE)
144     except socket.timeout:
145         pass
146     self.s.send(b"*OPC?\n")
147     self.s.settimeout(3)
148     try:
149         self.s.recv(BUFFERSIZE)
150     except socket.timeout:
151         pass
152
153     # Enviar al zera ordenes de configuraci n. Recibir datos?
154     def configurar_tensiones_evt(self):
155         """Funci n para configurar las tensiones primaria y secundaria del
156             patr n de ensayo. Este manda las ordenes de configuraci n al
157             Puente Zera.
158
159             Parametros:
160                 self -> clase PuenteZera
161
162             Atributos:
163                 orden -> Orden de configuraci n que se le env a al
164                 PUENTE ZERA
165
166             Excepciones:
167                 Socket TimeOut - Si la conexi n tarda m s de 0.2
168                 segundos en recibir la informaci n se pasa a la
169                 siguiente orden de configuraci n.
170
171             """
172         orden = "CONF:RAT:EVT " + self.EVTprim + "V/w3:" + self.EVTsec + "V
173             /w3\n"
174         self.s.send(b"CONF:RAT:EVT?\n")
175         self.s.settimeout(3)
176         try:
177             self.s.recv(BUFFERSIZE)
178         except socket.timeout:
179             pass
180         self.s.send(bytes(orden, "ascii"))
181         self.s.send(b"CONF:APPLY\n")
182         self.s.settimeout(3)
183         try:
184             self.s.recv(BUFFERSIZE)
185         except socket.timeout:
186             pass
187
188     # Enviar al zera ordenes de configuraci n. Recibir datos?
189     def configurar_frecuencia(self):

```

```

190     """Funci n para configurar la frecuencia de la se al . Este manda
191         las ordenes de configuraci n al Puente Zera.
192
193     Parametros:
194         self -> clase PuenteZera
195
196     Atributos:
197         orden -> Orden de configuraci n que se le env a al
198             PUENTE ZERA
199
200     Excepciones:
201         Socket TimeOut - Si la conexi n tarda m s de 0.2
202             segundos en recibir la informaci n se pasa a la
203             siguiente orden de configuraci n.
204
205     """
206     orden = "CONF:MEAS:SIGF " + self.frecuencia + "\n"
207     self.s.send(b"CONF:MEAS:SIGF?\n")
208     self.s.settimeout(3)
209     try:
210         self.s.recv(BUFFERSIZE)
211     except socket.timeout:
212         pass
213     self.s.send(bytes(orden, "ascii"))
214     self.s.send(b"CONF:APPLY\n")
215     self.s.settimeout(3)
216     try:
217         self.s.recv(BUFFERSIZE)
218     except socket.timeout:
219         pass
220     self.s.send(b"*OPC?\n")
221     self.s.settimeout(3)
222     try:
223         self.s.recv(BUFFERSIZE)
224     except socket.timeout:
225         pass
226
227 # Enviar al zera ordenes de configuraci n inicial. Recibir datos?
228 def conf_inicial(self):
229     """Funci n que lee del Puente Zera la configuraci n de la
230         frecuencia y las tensiones en el momento que se inicia.
231
232     Parametros:
233         self -> clase PuenteZera
234
235     Atributos:
236         orden -> Orden de configuraci n que se le env a al
237             PUENTE ZERA
238
239     Excepciones:
240         Socket TimeOut - Si la conexi n tarda m s de 0.5
241             segundos en recibir la informaci n se pasa a la
242             siguiente orden de configuraci n.
243
244     """
245     self.s.send(b"CONF:MEAS:SIGF?\n")
246     self.s.settimeout(3)
247     f = b""

```

```

240     try:
241         f = self.s.recv(BUFFERSIZE)
242     except socket.timeout:
243         pass
244     self.set_frecuencia(f.decode().rstrip("\nHz"))
245
246     self.s.send(b"CONF:RAT:EVT?\n")
247     self.s.settimeout(3)
248     evt = b""
249     try:
250         evt = self.s.recv(BUFFERSIZE)
251     except socket.timeout:
252         pass
253
254     evt_str = evt.decode().rstrip("\nV").replace("V", "").replace("/w3"
255         , "")
256     evt_list = evt_str.split(":")
257     if len(evt_list) == 2:
258         self.set_evt_prim(evt_list[0])
259         self.set_evt_sec(evt_list[1])
260     else:
261         self.set_n_prim(self.EVTprim)
262         self.set_n_sec(self.EVTsec)
263
264     self.s.send(b"CONF:RAT:N?\n")
265     self.s.settimeout(3)
266     n = b""
267     try:
268         n = self.s.recv(BUFFERSIZE)
269     except socket.timeout:
270         pass
271     n_str = n.decode().rstrip("\nV").replace("V", "").replace("/w3", ""
272         )
273     n_list = n_str.split(":")
274     if len(evt_list) == 2:
275         self.set_n_prim(n_list[0])
276         self.set_n_sec(n_list[1])
277     else:
278         self.set_n_prim(self.Nprim)
279         self.set_n_sec(self.Nsec)
280
281     # Enviar al zera orden de leer. Recibir datos?
282     def leer(self):
283         """Funci n que manda la orden de lectura al Puente Zera.
284
285         ste responde con un String de datos el cual modificamos y
286         guardamos en atributos de la clase mediante la funci n
287         datos.
288
289         String: Frequency;VectorN;VectorX;Loadpoint;Loadpoint1;Error1;
290         Error2;Angle;RCF
291
292         Parametros:
293         self -> clase PuenteZera
294         """
295     try:
296         self.s.send(b"READ?\n")
297         # self.s.settimeout(None)

```

```

294         self.s.settimeout(10)
295         data_read = self.s.recv(BUFSIZE)
296         self.datos(data_read)
297     except socket.timeout as e:
298         print(e)
299
300     def datos(self, dr):
301         """Funci n que recoge el bit String que nos devuelve el Puente
302             Zera y guarda los datos.
303
304             Primero lo convierte de bits a String. Despu s crea una lista con
305             estos valores y los modifica a nuestro gusto. (N mero de
306             decimales; Desfase en minutos)
307
308             frecuencia=dr_list[0]
309             VectorN=dr_list[1,2]
310             VectorX=dr_list[3,4]
311             LoadPoint=dr_list[5]
312             LoadPoint1=dr_list[6]
313             Error1=dr_list[7]
314             Error2=dr_list[8]
315             Angle=dr_list[9]
316             RCF=dr_list[10]
317
318             Parametros:
319                 self -> clase PuenteZera
320                 dr -> bytes que recibimos en la funci n leer.
321
322         """
323         dr_str = dr.decode() # Decodificaci n de la informaci n que nos
324                             # da el Puente Zera bits -> String
325         dr_list = dr_str.split(";")
326         a = float(dr_list[9]) * 3437.74677 # desfase: radianes -> minutos
327         a = a.__round__(0)
328         self.set_frecuencia2(dr_list[0])
329         b = float(dr_list[7]).__round__(2)
330         self.set_error_modulo(b.__str__())
331         self.set_error_angulo(a.__str__())
332         self.set_lpn(dr_list[6])
333         c = float(dr_list[5]).__round__(2)
334         self.set_lpn(c.__str__())
335         self.set_rcf(dr_list[10])
336         d = float(self.LPN) * float(self.Nprim) / 100000 / math.sqrt(3)
337         d = d.__round__(2)
338         self.set_u(d.__str__())

```

## B. Hipotronics Klasea

```

1 import socket
2
3 # VARIABLES GLOBALES A futuro deben tomar los valores de un fichero de
  configuraci n
4
5 BUFFERSIZE = 1024 # tama o del buffer de datos para el uso del socket
6
7 START_BYTE = 0x3A # El byte de comienzo. Com n para todos los comandos.
8 END_BYTE = 0x0D # El byte de final. Com n para todos los comandos.
9
10 TRIP_CURRENT = "30" # Intensidad m xima por seguridad. No cambia.
11
12 TCP_IP = "192.168.0.5" # IP
13 TCP_PORT = 4001 # N mero de puerto
14
15
16 class Hipotronics(object):
17     """Clase para uso y control del dispositivo HIPOTRONICS 700-DI.
18
19     Se utiliza la libreria socket para establecer conexi on TCP/IP con el
20     dispositivo y as mandar ordenes de cambio de configuraci n y de
21     lectura de medidas.
22
23     Es necesario poner el PC y el dispositivo en el mismo area de IP con
24     solo el ltimo digito cambiado. Los detalles de una conexi n
25     optima est n recogidos en el document 'Sensores_700-DI_Protocol' el
26     documento de protocolos del dispositivo.
27
28     Todos los comandos est n recogidos en el documento previamente
29     mencionado. Los comandos se basan en mandar n meros hexadecimales
30     en el orden correcto y siguiendo el manjua para cada comando. El
31     orden es el siguiente:
32     START BYTE—> LENGTH —> FUNCTION CODE —> DATA —> CHECKSUM —>
33     END BYTE
34
35     NOTAS:
36     1.—Start byte siempre es 0x3A y el End Byte 0x0D
37     2.—Length hace referencia a la largura de DATA, si no se manda
38     ning n dato el length ser 0.
39     3.— El checksum es la multiplicaci n en hexadecimal de todos
40     los datos anteriores.
41
42     """
43
44     # Inicializar valores y conectar socket
45     def __init__(self):
46         """Funci on por defecto de inicializaci on de objeto dentro de la
47         clase. Se les asignan valores de inicializaci n a los atributos
48         del objeto.
49
50         Parametros:
51             self—> clase Hipotronics
52
53         Atributos:

```



```

43     self.connected — Variable que nos indica si el socket est
         conectado. 1—>conectado 0—>Desconectado
44
45     self.ramp1 — Primera rampa de subida en kV/s
46
47     self.ramp2 — Segunda rampa de bajada? en kV/s
48
49     self.ramp3 — Tercera rampa de bajada? en kV/s
50
51     self.ramp4 — Cuarta rampa de bajada? en kV/s
52
53     self.ramp5 — Quinta rampa de bajada? en kV/s
54
55     self.targetVoltage1 — Primer punto de medida en kV
56
57     self.targetVoltage2 — Segundo punto de medida en kV
58
59     self.targetVoltage3 — Tercer punto de medida en kV
60
61     self.targetVoltage4 — Cuarto punto de medida en kV
62
63     self.targetVoltage5 — Quinto punto de medida en kV
64
65     self.dwell1 — Tiempo de espera en el primer punto en segundos
66
67     self.dwell2 — Tiempo de espera en el segundo punto en segundos
68
69     self.dwell3 — Tiempo de espera en el tercer punto en segundos
70
71     self.dwell4 — Tiempo de espera en el cuarto punto en segundos
72
73     self.dwell5 — Tiempo de espera en el quinto punto en segundos
74
75     self.tripVoltage — Tensi n m xima por seguridad.
76
77     self.MeasurementData — Dato que nos manda el HIPOTRONICS sobre
         la medida actual.
78
79     """
80     # Inicializaci n de atributos definidos:
81     self.connected = None
82     self.ramp1 = "1.5"
83     self.ramp2 = "1.5"
84     self.ramp3 = "1.5"
85     self.ramp4 = "1.5"
86     self.ramp5 = "1.5"
87     self.targetVoltage1 = "0"
88     self.targetVoltage2 = "0"
89     self.targetVoltage3 = "0"
90     self.targetVoltage4 = "0"
91     self.targetVoltage5 = "0"
92     self.dwell1 = "5"
93     self.dwell2 = "5"
94     self.dwell3 = "5"
95     self.dwell4 = "5"
96     self.dwell5 = "5"
97     self.tripVoltage = (float(self.targetVoltage1) + 1).__str__()
98     self.MeasurementData = 0

```

```

99
100 def conectar_hipotronics(self): # Hipotronicis en la subred 1
101     """Funci n que abre el socket y la conexi n con el HIPOTRONICS.
102
103     Parametros:
104         self → clase Hipotronics
105
106         tcp_ip → IP del Hipotronics
107
108         tcp_port → n mero de puerto del Hipotronics
109
110     Atributos:
111         self.s — socket de conexi n TCP/IP
112     Excepciones:
113         Socket Timeout — Si el socket tarda m s de 3 segundos en
            conectarse el programa dejara de intentarlo y se pondr
            atributo self.connected a 0 indicando que no se ha
            realizado la conexi n.
114
115     """
116     self.s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
117     self.s.settimeout(3)
118     try:
119         self.s.connect((TCP_IP, TCP_PORT))
120         self.connected = 1
121     except socket.timeout:
122         self.connected = 0
123
124 def desconectar_hipotronics(self):
125     if self.connected == 1:
126         self.s.close()
127         self.connected = 0
128
129 # Envia datos a H. Recibe ?
130 def set_steps(self):
131     """ Funci n que le env a la orden de definir la rampa en modo
132         autom tico al Hipotronics.
133
134     Parametros:
135         self → clase Hipotronics
136
137     Atributos:
138         data → string de los datos que se le van a enviar en el
139             comando al HIPOTRONICS.
140
141         length → largura de data (len(data))
142
143         function_code → El c digo correspondiente a la funci n
144
145         checksum → Multiplicaci n l gica (OR) de todos los bits
146             anteriormente mandados
147
148         b → el byte string final que se manda mediante el comando
149             con todos los atributos anteriores en el orden correcto
150
151     Excepciones:
152         Socket Timeout — Si el socket tarda m s de 1 segundo en
153             conectarse el programa dejara de intentarlo.
154
155     """

```

```

149
150     """
151     data = self.ramp1 + "*" + self.targetVoltage1 + "*" + self.dwell1 +
        "&" + self.ramp2 + "*" + self.targetVoltage2 + "*" + self.
        dwell2 + "&" + self.ramp3 + "*" + self.targetVoltage3 + "*" +
        self.dwell3 + "&" + self.ramp4 + "*" + self.targetVoltage4 + "*"
        + self.dwell4 + "&" + self.ramp5 + "*" + self.targetVoltage5 +
        "*" + self.dwell5
152     length = len(data)
153     function_code = 0x10
154     checksum = START_BYTE ^ length ^ function_code
155     for i in range(0, len(data)):
156         checksum ^= ord(data[i])
157     b = START_BYTE.to_bytes(1, "little").__add__(length.to_bytes(1, "
        little")).__add__(
158         function_code.to_bytes(1, "little").__add__(data.encode("ASCII"
        )).__add__(
159         checksum.to_bytes(1, "little").__add__(END_BYTE.to_bytes(1,
        "little"))))))
160     self.s.send(b)
161     self.s.settimeout(1)
162     try:
163         self.s.recv(BUFFERSIZE)
164     except socket.timeout:
165         pass
166
167     # Envia datos a H. Recibe ?
168     def set_test_mode_auto(self):
169         """ Funci n que configura el Hipotronics en modo AUTOMTICO.
170
171         Parametros:
172             self -> clase Hipotronics
173
174         Atributos:
175             Data -> string de los datos que se le van a embiar en el
                comando al HIPOTRONICS.
176
177             length -> largura de Data (len(Data))
178
179             function_code -> El c digo correspondiente a la funci n
180
181             checksum -> Multiplicaci n l gica (OR) de todos los bits
                anteriormente mandados
182
183             b -> el byte string final que se manda mediante el comando
                con todos los atributos anteriores en el orden correcto
184
185         Excepciones:
186             Socket TimeOut - Si el socket tarda m s de 1 segundo en
                conectarse el programa dejara de intentarlo.
187
188
189     """
190     length = 0x01
191     function_code = 0x15
192     data = 0x32
193     checksum = START_BYTE ^ length ^ function_code ^ data

```

```

194     b = START_BYTE.to_bytes(1, "little").__add__(length.to_bytes(1, "
195         little").__add__(
196             function_code.to_bytes(1, "little").__add__(data.to_bytes(1, "
197                 little").__add__(
198                     checksum.to_bytes(1, "little").__add__(END_BYTE.to_bytes(1,
199                         "little"))))))))
200     self.s.send(b)
201     self.s.settimeout(1)
202     try:
203         self.s.recv(BUFSIZE)
204     except socket.timeout:
205         pass
206
207 # Envia datos a H. Recibe ? Diferencia con set_test_mode_auto
208 # ?
209
210 def set_test_mode_manual(self):
211     """ Funci n que configura el Hipotronics en modo MANUAL.
212
213     Parametros:
214         self -> clase Hipotronics
215
216     Atributos:
217         data -> string de los datos que se le van a embiar en el
218             comando al HIPOTRONICS.
219
220         length -> largura de data (len(data))
221
222         function_code -> El c digo correspondiente a la funci n
223
224         checksum -> Multiplicaci n l gica (OR) de todos los bits
225             anteriormente mandados
226
227         b -> el byte string final que se manda mediante el comando
228             con todos los atributos anteriores en el orden correcto
229
230     Excepciones:
231         Socket TimeOut - Si el socket tarda m s de 1 segundo en
232             conectarse el programa dejara de intentarlo.
233
234     """
235     length = 0x01
236     function_code = 0x15
237     data = 0x31
238     checksum = START_BYTE ^ length ^ function_code ^ data
239     b = START_BYTE.to_bytes(1, "little").__add__(length.to_bytes(1, "
240         little").__add__(
241             function_code.to_bytes(1, "little").__add__(data.to_bytes(1, "
242                 little").__add__(
243                     checksum.to_bytes(1, "little").__add__(END_BYTE.to_bytes(1,
244                         "little"))))))))
245     self.s.send(b)
246     self.s.settimeout(1)
247     try:
248         self.s.recv(BUFSIZE)
249     except socket.timeout:
250         pass
251
252 def get_test_mode(self):

```

```

241     """ Funci n que nos dice en que modo est  el HIPOTRONICS.
242
243     Devuelve un n mero hexadecimal dependiendo del modo — Home: x10;
        Manual: x11; Auto: x12.
244
245     Parametros:
246         self —> clase Hipotronics
247
248     Atributos:
249         Data —> string de los datos que se le van a embiar en el
        comando al HIPOTRONICS.
250
251         length —> largura de Data (len(Data))
252
253         function_code —> El c digo correspondiente a la funci n
254
255         checksum —> Multiplicaci n l gica (OR) de todos los bits
        anteriormente mandados
256
257         b —> el byte string final que se manda mediante el comando
        con todos los atributos anteriores en el orden correcto
258
259     Parametros de salida:
260         test_mode —> informaci n en forma de bits que nos dice en
        que modo est  el HIPOTRONICS.
261
262     Excepciones:
263         Socket TimeOut — Si el socket tarda m s de 1 segundo en
        conectarse el programa dejara de intentarlo.
264
265     """
266     length = 0x00
267     function_code = 0x2B
268     checksum = 0x11
269     b = START_BYTE.to_bytes(1, "little").__add__(length.to_bytes(1, "
        little").__add__(
270         function_code.to_bytes(1, "little").__add__(
271         checksum.to_bytes(1, "little").__add__(END_BYTE.to_bytes(1,
        "little")))))
272     self.s.send(b)
273     self.s.settimeout(1)
274     datos = ""
275     try:
276         datos = self.s.recv(BUFFERSIZE)
277     except socket.timeout:
278         pass
279
280     test_mode = datos.removeprefix(b":\x06\x010000100\x0c\r:\x05
        (0146.6\x0c\r:\x01+02")
281
282     return test_mode
283
284     def get_steps(self):
285         """ Funci n que nos da los datos de las rampas.
286
287         Devuelve los datos de la rampa grabados en ese instante en el
        HIPOTRONICS.
288

```

```

289     Parametros:
290         self → clase Hipotronics
291
292     Atributos:
293         Data → string de los datos que se le van a emviar en el
                comando al HIPOTRONICS.
294
295         length → largura de Data (len(Data))
296
297         function_code → El c digo correspondiente a la funci n
298
299         checksum → Multiplicaci n l gica (OR) de todos los bits
                anteriormente mandados
300
301         b → el byte string final que se manda mediante el comando
                con todos los atributos anteriores en el orden correcto
302
303     Parametros de salida:
304         steps → datos en bits con la siguiente forma: rampa1*
                tensi n1*tiempo1&rampa2*tensi n2*tiempo2&...&rampaN*
                tensi nN*tiempoN
305
306     Excepciones:
307         Socket Timeout — Si el socket tarda m s de 1 segundo en
                conectarse el programa dejara de intentarlo.
308
309     """
310     length = 0x00
311     function_code = 0x11
312     checksum = 0x2B
313     b = START_BYTE.to_bytes(1, "little").__add__(length.to_bytes(1, "
        little").__add__(
314         function_code.to_bytes(1, "little").__add__(
315             checksum.to_bytes(1, "little").__add__(END_BYTE.to_bytes(1,
                "little")))))
316     self.s.send(b)
317     self.s.settimeout(1)
318     steps = ""
319     try:
320         steps = self.s.recv(BUFFERSIZE)
321     except socket.timeout:
322         pass
323     steps = steps.removeprefix(b":\x06\x010000100\x0c\r:\x05(0146.6\x0c
        \r:\x11\x110")
324     return steps
325
326     def get_serial_number(self):
327         """ Funci n que nos devuelve el n mero de serie del Hipotronics.
328
329         Parametros:
330             self → clase Hipotronics
331
332         Atributos:
333             Data → string de los datos que se le van a emviar en el
                comando al HIPOTRONICS.
334
335             length → largura de Data (len(Data))
336

```

```

337         function_code → El código correspondiente a la función
338
339         checksum → Multiplicación lógica (OR) de todos los bits
           anteriormente mandados
340
341         b → el byte string final que se manda mediante el comando
           con todos los atributos anteriores en el orden correcto
342
343     Parametros de salida:
344         serial_number → datos en bits que incluyen el número de
           serie
345     Excepciones:
346         Socket Timeout — Si el socket tarda más de 0.2 segundo en
           conectarse el programa dejara de intentarlo.
347
348
349     """
350     length = 0x00
351     function_code = 0x03
352     checksum = 0x39
353     b = START_BYTE.to_bytes(1, "little").__add__(length.to_bytes(1, "
           little").__add__(
354         function_code.to_bytes(1, "little").__add__(
355             checksum.to_bytes(1, "little").__add__(END_BYTE.to_bytes(1,
           "little")))))
356     self.s.send(b)
357     self.s.settimeout(0.2)
358     serial_number = ''
359     try:
360         serial_number = self.s.recv(BUFSIZE)
361     except socket.timeout:
362         pass
363     return serial_number
364
365 def set_hv_on(self):
366     """ Función para comenzar el proceso.
367
368     Parametros:
369         self → clase Hipotronics
370
371     Atributos:
372         data → string de los datos que se le van a enviar en el
           comando al HIPOTRONICS.
373
374         length → largura de data (len(data))
375
376         function_code → El código correspondiente a la función
377
378         checksum → Multiplicación lógica (OR) de todos los bits
           anteriormente mandados
379
380         b → el byte string final que se manda mediante el comando
           con todos los atributos anteriores en el orden correcto
381
382     Excepciones:
383         Socket Timeout — Si el socket tarda más de 1 segundo en
           conectarse el programa dejara de intentarlo.
384

```

```

385     """
386     length = 0x01
387     function_code = 0x2F
388     data = 0x31
389     checksum = 0x25
390     b = START_BYTE.to_bytes(1, "little").__add__(length.to_bytes(1, "
391         little").__add__(
392             function_code.to_bytes(1, "little").__add__(data.to_bytes(1, "
393                 little").__add__(
394                     checksum.to_bytes(1, "little").__add__(END_BYTE.to_bytes(1,
395                         "little"))))))))
396     self.s.send(b)
397     self.s.settimeout(1)
398     try:
399         self.s.recv(BUFFERSIZE)
400     except socket.timeout:
401         pass
402
403 def set_hv_off(self):
404     """ Funci n para parar el proceso sin que haya acabado.
405
406     Parametros:
407         self —> clase Hipotronics
408
409     Atributos:
410         data —> string de los datos que se le van a embiar en el
411             comando al HIPOTRONICS.
412
413         length —> largura de data (len(data))
414
415         function_code —> El c digo correspondiente a la funci n
416
417         checksum —> Multiplicaci n l gica (OR) de todos los bits
418             anteriormente mandados
419
420         b —> el byte string final que se manda mediante el comando
421             con todos los atributos anteriores en el orden correcto
422
423     Excepciones:
424         Socket TimeOut — Si el socket tarda m s de 1 segundo en
425             conectarse el programa dejara de intentarlo.
426
427     """
428     length = 0x01
429     function_code = 0x2F
430     data = 0x30
431     checksum = 0x24
432     b = START_BYTE.to_bytes(1, "little").__add__(length.to_bytes(1, "
433         little").__add__(
434             function_code.to_bytes(1, "little").__add__(data.to_bytes(1, "
435                 little").__add__(
436                     checksum.to_bytes(1, "little").__add__(END_BYTE.to_bytes(1,
437                         "little"))))))))
438     self.s.send(b)
439     self.s.settimeout(1)
440     try:
441         self.s.recv(BUFFERSIZE)
442     except socket.timeout:

```



```

433         pass
434
435     def send_measurement_data(self):
436         """ Funci n que nos da la informaci n de la medida realizada por
437             el HIPOTRONICS.
438
439             Con este dato actualizamos el atributo self.measurement_data.
440
441             Nota: el bit stream que nos manda el HIPOTRONICS no es siempre
442                 exactamente igual, por ello hay que manipular los datos
443                 recibidos para quedarnos siempre con el que nos interesa.
444
445         Parametros:
446             self → clase Hipotronics
447
448         Atributos:
449             Data → string de los datos que se le van a embiar en el
450                 comando al HIPOTRONICS.
451
452             length → largura de Data (len(Data))
453
454             function_code → El c digo correspondiente a la funci n
455
456             checksum → Multiplicaci n l gica (OR) de todos los bits
457                 anteriormente mandados
458
459             b → el byte string final que se manda mediante el comando
460                 con todos los atributos anteriores en el orden correcto
461
462         Excepciones:
463             Socket TimeOut — Si el socket tarda m s de 0.2 segundo en
464                 conectarse el programa dejara de intentarlo.
465
466         """
467         length = 0x00
468         function_code = 0x23
469         checksum = 0x19
470         b = START_BYTE.to_bytes(1, "little").__add__(length.to_bytes(1, "
471             little").__add__(
472                 function_code.to_bytes(1, "little").__add__(
473                     checksum.to_bytes(1, "little").__add__(END_BYTE.to_bytes(1,
474                         "little")))))
475         self.s.send(b)
476         self.s.settimeout(1)
477         steps = ""
478         try:
479             steps = self.s.recv(BUFFERSIZE)
480         except socket.timeout:
481             pass
482         if steps.__contains__(b"#"):
483             measurement_data = steps.split(b"#")
484             measurement_data = measurement_data[1].split(b"*.")
485             if measurement_data[0] == b":":
486                 self.MeasurementData = measurement_data[2]
487             else:
488                 self.MeasurementData = measurement_data[1]

```

```

482 def set_trip_voltage(self):
483     """ Funci n que cambia el Target Voltage del modo.
484
485     Parametros:
486         self → clase Hipotronics
487
488     Atributos:
489         data → string de los datos que se le van a embiar en el
490             comando al HIPOTRONICS.
491
492         length → largura de data (len(data))
493
494         function_code → El c digo correspondiente a la funci n
495
496         Checksum → Multiplicaci n l gica (OR) de todos los bits
497             anteriormente mandados
498
499         b → el byte string final que se manda mediante el comando
500             con todos los atributos anteriores en el orden correcto
501
502     Excepciones:
503         Socket TimeOut — Si el socket tarda m s de 1 segundo en
504             conectarse el programa dejara de intentarlo.
505
506     """
507     data = self.tripVoltage
508     length = len(data)
509     function_code = 0x29
510     checksum = START_BYTE ^ length ^ function_code
511     for i in range(0, len(data)):
512         checksum ^= ord(data[i])
513     b = START_BYTE.to_bytes(1, "little").__add__(length.to_bytes(1, "
514         little").__add__(
515             function_code.to_bytes(1, "little").__add__(data.encode("ASCII"
516                 ).__add__(
517                 checksum.to_bytes(1, "little").__add__(END_BYTE.to_bytes(1,
518                     "little"))))))))
519     self.s.send(b)
520
521     self.s.settimeout(2)
522     try:
523         self.s.recv(BUFFERSIZE)
524     except socket.timeout:
525         pass
526
527 def set_trip_current(self):
528     """ Funci n que cambia el Target Current del modo.
529
530     Parametros:
531         self → clase Hipotronics
532
533     Atributos:
534         data → string de los datos que se le van a embiar en el
535             comando al HIPOTRONICS.
536
537         length → largura de data (len(data))
538
539         function_code → El c digo correspondiente a la funci n

```

```

532
533         checksum —> Multiplicaci n l gica (OR) de todos los bits
                    anteriormente mandados
534
535         b —> el byte string final que se manda mediante el comando
                    con todos los atributos anteriores en el orden correcto
536
537     Excepciones:
538         Socket TimeOut — Si el socket tarda m s de 1 segundo en
                    conectarse el programa dejara de intentarlo.
539
540     """
541     data = TRIP_CURRENT
542     length = len(data)
543     function_code = 0x2A
544     checksum = START_BYTE ^ length ^ function_code
545     for i in range(0, len(data)):
546         checksum ^= ord(data[i])
547     b = START_BYTE.to_bytes(1, "little").__add__(length.to_bytes(1, "
548         little").__add__(
549         function_code.to_bytes(1, "little").__add__(data.encode("ASCII"
550         ).__add__(
551         checksum.to_bytes(1, "little").__add__(END_BYTE.to_bytes(1,
552         "little"))))))
553     self.s.send(b)
554     self.s.settimeout(1)
555     try:
556         self.s.recv(BUFFERSIZE)
557     except socket.timeout:
558         pass
559
560     def get_ping_info(self):
561         """ Funci n que nos informaci n del estado del HIPOTRONICS.
562
563         Parametros:
564             self—> clase Hipotronics
565
566         Atributos:
567             Data—> string de los datos que se le van a embiar en el
568                 comando al HIPOTRONICS.
569
570             length —> largura de Data (len(Data))
571
572             function_code —> El c digo correspondiente a la funci n
573
574             checksum —> Multiplicaci n l gica (OR) de todos los bits
575                 anteriormente mandados
576
577             b —> el byte string final que se manda mediante el comando
578                 con todos los atributos anteriores en el orden correcto
579
580         Parametros de salida:
581             ping—> bytes con informaci n actual sobre el dispositivo
582
583         Excepciones:
584             Socket TimeOut — Si el socket tarda m s de 0.2 segundo en
585                 conectarse el programa dejara de intentarlo.
586
587     """

```

```
580     length = 0x00
581     function_code = 0x01
582     checksum = 0x3B
583     b = START_BYTE.to_bytes(1, "little").__add__(length.to_bytes(1, "
584         little").__add__(
585             function_code.to_bytes(1, "little").__add__(
586                 checksum.to_bytes(1, "little").__add__(END_BYTE.to_bytes(1,
587                     "little")))))
588     self.s.send(b)
589     self.s.settimeout(0.2)
590     ping = ""
591     try:
592         ping = self.s.recv(BUFFERSIZE)
593     except socket.timeout:
594         pass
595     return ping
```

## C. ComunicacionWebService Klasea

```

1 from zeep import Client , Settings
2 import xmldict
3
4 # VARIABLES GLOBALES A futuro deben tomar los valores de un fichero de
   configuraci n
5 ENSAYO_DDPP = 25 # Identificador del ensayo DDP
6 ENSAYO_PRECISION = 3 # Identificador del ensayo PRECISION
7
8
9 class ComunicacionWebService(object):
10     """Clase para la comunicaci n con el web service.
11
12     La clase nos permite acceder al web service 'https://saio.jakiten.com/
   Ensayos/WebServices/wsPuntosPrecision.asmx?wsdl' y mediante usuario
   y password recibir y mandar datos de ensayo.
13
14     Para la obtenci n de datos se accede a dos bases de datos: SAIO y DIS3
   . Estos re reciben en formato XML y se guardan en nuevos variables
   para su uso.
15
16     Los datos obtenidos de ensayos se mandan a SAIO directamente en formato
   str pero con forma de XML.
17     """
18
19     # Inicializar variables
20     def __init__(self):
21         """Funcion por defecto de inicializacion de objeto dentro de la
   clase. Se les asignan valores de inicializaci n a los atributos
   del objeto.
22
23         Parametros:
24             self : clase ComunicacionWebService
25
26         Atributos:
27
28             Atributos SAIO:
29                 self.ID_DATOS_ENSAYO — parametro de ensayo referente
   al tipo de ensayo. 3 —>Precisi n 25 —>DDPP
30
31                 self.OV — Orden de venta
32
33                 self.NUM_APARATO — N mero de aparato
34
35                 self.ENSAYO_ETAPA_FAB — Etapa de fabricaci n
36
37                 self.DESCRIPCION_TANDA — Descripci n de la tanada (ej
   : '1er ensayo')
38
39                 self.NUM_TANDA — N mero de tanda
40
41             Valores ensayo de precisi n:
42
43                 self.errorModulo1 — Valor del error en m dulo en
   el primer punto para ensayo de precisi n.
44

```

45 *self.errorModulo2* — Valor del error en módulo en  
el segundo punto para ensayo de precisión.

46

47 *self.errorModulo3* — Valor del error en módulo en  
el tercer punto para ensayo de precisión.

48

49 *self.errorModulo4* — Valor del error en módulo en  
el cuarto punto para ensayo de precisión.

50

51 *self.errorModulo5* — Valor del error en módulo en  
el quinto punto para ensayo de precisión.

52

53 *self.errorAngulo1* — Valor del error en ángulo en  
el primer punto para ensayo de precisión.

54

55 *self.errorAngulo2* — Valor del error en ángulo en  
el segundo punto para ensayo de precisión.

56

57 *self.errorAngulo3* — Valor del error en ángulo en  
el tercer punto para ensayo de precisión.

58

59 *self.errorAngulo4* — Valor del error en ángulo en  
el cuarto punto para ensayo de precisión.

60

61 *self.errorAngulo5* — Valor del error en ángulo en  
el quinto punto para ensayo de precisión.

62

63 *Valores ensayo descargas parciales:*

64

65 *self.valor\_ddpp\_1* — Valor ensayo descargas  
parciales en el primer punto.

66

67 *self.valor\_ddpp\_2* — Valor ensayo descargas  
parciales en el segundo punto.

68

69 *self.valor\_ddpp\_3* — Valor ensayo descargas  
parciales en el tercer punto.

70

71 *self.valor\_ddpp\_4* — Valor ensayo descargas  
parciales en el cuarto punto.

72

73 *self.valor\_ddpp\_5* — Valor ensayo descargas  
parciales en el quinto punto.

74

75

76 *Atributos DIS3:*

77

78 *self.cantidad\_aparatos* — Cantidad de aparatos a  
ensayar

79

80 *self.carga\_capac* — Carga capacitiva

81

82 *self.carga\_resist* — Carga resistiva

83

84 *self.CLASE* — Clase del dispositivo

85

86 *self.factor\_tension* — Dato de DIS3

87

88 *self.nombre\_cliente* — *Cliente*  
89  
90 *self.norma* — *NORMA*  
91  
92 *self.orden\_fab* — *Orden de fabricaci n*  
93  
94 *self.OV\_IN\_OUT* — *Orden de venta*  
95  
96 *self.temp\_max* — *Temperatura m xima*  
97  
98 *self.temp\_min* — *Temperatura m nima*  
99  
100 *Configuraci n ensayo de precisi n*  
101  
102 *self.num\_t\_1\_sec* — *EVTs*  
103  
104 *self.num\_t\_prim* — *EVTp*  
105  
106 *self.frecuencia* — *frecuencia de ensayo*  
107  
108 *Descargas parciales:*  
109  
110 *self.lim\_1\_ddpp* — *limite aceptable en el primer*  
111 *punto de medida de descargas parciales. En pC.*  
112  
113 *self.lim\_2\_ddpp* — *limite aceptable en el segundo*  
114 *punto de medida de descargas parciales. En pC.*  
115  
116 *self.lim\_3\_ddpp* — *limite aceptable en el tercer*  
117 *punto de medida de descargas parciales. En pC.*  
118  
119 *self.lim\_4\_ddpp* — *limite aceptable en el cuarto*  
120 *punto de medida de descargas parciales. En pC.*  
121  
122 *self.lim\_5\_ddpp* — *limite aceptable en el quinto*  
123 *punto de medida de descargas parciales. En pC.*  
124  
125 *self.pto\_1\_ddp* — *primer punto de medida de*  
126 *descargas parciales. En kV.*  
127  
128 *self.pto\_2\_ddp* — *primer punto de medida de*  
129 *descargas parciales. En kV.*  
130  
131 *self.pto\_3\_ddp* — *primer punto de medida de*  
132 *descargas parciales. En kV.*  
133  
134 *self.pto\_4\_ddp* — *primer punto de medida de*  
135 *descargas parciales. En kV.*  
136  
137 *self.pto\_5\_ddp* — *primer punto de medida de*  
138 *descargas parciales. En kV.*  
139  
140 *Precisi n:*  
141  
142 *self.lim\_angulo\_1\_positivo* — *limite positivo*  
143 *aceptable en desfase en el primer punto de*  
144 *medida de precisi n. En minutos.*  
145

134 *self.lim\_angulo\_2\_positivo* — limite positivo  
aceptable en desfase en el segundo punto de  
medida de precisión. En minutos.

135  
136 *self.lim\_angulo\_3\_positivo* — limite positivo  
aceptable en desfase en el tercer punto de  
medida de precisión. En minutos.

137  
138 *self.lim\_angulo\_4\_positivo* — limite positivo  
aceptable en desfase en el cuarto punto de  
medida de precisión. En minutos.

139  
140 *self.lim\_angulo\_5\_positivo* — limite positivo  
aceptable en desfase en el quinto punto de  
medida de precisión. En minutos.

141  
142 *self.lim\_angulo\_1\_negativo* — limite negativo  
aceptable en desfase en el primer punto de  
medida de precisión. En minutos.

143  
144 *self.lim\_angulo\_2\_negativo* — limite negativo  
aceptable en desfase en el segundo punto de  
medida de precisión. En minutos.

145  
146 *self.lim\_angulo\_3\_negativo* — limite negativo  
aceptable en desfase en el tercer punto de  
medida de precisión. En minutos.

147  
148 *self.lim\_angulo\_4\_negativo* — limite negativo  
aceptable en desfase en el cuarto punto de  
medida de precisión. En minutos.

149  
150 *self.lim\_angulo\_5\_negativo* — limite negativo  
aceptable en desfase en el quinto punto de  
medida de precisión. En minutos.

151  
152 *self.lim\_modulo\_negativo\_1* — limite negativo  
aceptable en modulo en el primer punto de medida  
de precisión. En %.

153  
154 *self.lim\_modulo\_negativo\_2* — limite negativo  
aceptable en modulo en el segundo punto de  
medida de precisión. En %.

155  
156 *self.lim\_modulo\_negativo\_3* — limite negativo  
aceptable en modulo en el tercer punto de medida  
de precisión. En %.

157  
158 *self.lim\_modulo\_negativo\_4* — limite negativo  
aceptable en modulo en el cuarto punto de medida  
de precisión. En %.

159  
160 *self.lim\_modulo\_negativo\_5* — limite negativo  
aceptable en modulo en el quinto punto de medida  
de precisión. En %.

161  
162 *self.lim\_modulo\_positivo\_1* — limite positivo  
aceptable en modulo en el primer punto de medida



```

163         de precisi n. En %.
164     self.lim_modulo_positivo_2 — limite positivo
165         aceptable en modulo en el segundo punto de
166         medida de precisi n. En %.
167
168     self.lim_modulo_positivo_3 — limite positivo
169         aceptable en modulo en el tercer punto de medida
170         de precisi n. En %.
171
172     self.lim_modulo_positivo_4 — limite positivo
173         aceptable en modulo en el cuarto punto de medida
174         de precisi n. En %.
175
176     self.lim_modulo_positivo_5 — limite positivo
177         aceptable en modulo en el quinto punto de medida
178         de precisi n. En %.
179
180     self.nivel_1_precision — primer punto de medida de
181         precisi n. En %.
182
183     self.nivel_2_precision — segundo punto de medida
184         de precisi n. En %.
185
186     self.nivel_3_precision — tercer punto de medida de
187         precisi n. En %.
188
189     self.nivel_4_precision — cuarto punto de medida de
190         precisi n. En %.
191
192     self.nivel_5_precision — quinto punto de medida de
193         precisi n. En %.
194
195     Variables de respuesta:
196
197     self.responseDIS3 — respuesta del DIS3
198
199     self.response — respuesta del SAIO
200
201     Variables del web service:
202
203     self.prueba — direcci n web del web service de prueba
204
205     self.wsdl — direcci n web del web service
206
207     self.client — objeto de la clase Client. Se utiliza para
208         la conexi n con el web service
209
210     self.header_value — cabecera de conexi n con web service
211         que contiene el usuario y la contrase a
212
213     self.identificador_nueva_tanda — Identificador de nueva
214         tanda
215
216     """
217     # Variables Saio
218     self.ID_DATOS_ENSAYO = None
219     self.OV = None

```

```
204     self.NUM_APARATO = None
205     self.ENSAYO_ETAPA_FAB = None
206     self.DESCRIPCION_TANDA = None
207     self.NUM_TANDA = None
208
209     # Definición valores del ensayo precisión
210     self.errorModulo1 = None
211     self.errorModulo2 = None
212     self.errorModulo3 = None
213     self.errorModulo4 = None
214     self.errorModulo5 = None
215
216     self.errorAngulo1 = None
217     self.errorAngulo2 = None
218     self.errorAngulo3 = None
219     self.errorAngulo4 = None
220     self.errorAngulo5 = None
221
222     # Definición valores del ensayo de descargas parciales
223     self.valor_ddpp_1 = None
224     self.valor_ddpp_2 = None
225     self.valor_ddpp_3 = None
226     self.valor_ddpp_4 = None
227     self.valor_ddpp_5 = None
228
229     # VARIABLES DIS3
230     self.cantidad_aparatos = None
231     self.carga_capac = None
232     self.carga_resist = None
233     self.CLASE = None
234     self.factor_tension = None
235     self.modelo = None
236     self.nombre_cliente = None
237     self.norma = None
238     self.orden_fab = None
239     self.OV_IN_OUT = None
240     self.temp_max = None
241     self.temp_min = None
242
243     # Configuración ensayo precisión.
244     self.num_t_1_sec = None
245     self.num_t_prim = None
246     self.frecuencia = None
247
248     # DDPP
249     self.lim_1_ddpp = None
250     self.lim_2_ddpp = None
251     self.lim_3_ddpp = None
252     self.lim_4_ddpp = None
253     self.lim_5_ddpp = None
254
255     self.pto_1_ddpp = None
256     self.pto_2_ddpp = None
257     self.pto_3_ddpp = None
258     self.pto_4_ddpp = None
259     self.pto_5_ddpp = None
260
261     # Precisión
```

```

262     self.lim_angulo_1_positivo = None
263     self.lim_angulo_2_positivo = None
264     self.lim_angulo_3_positivo = None
265     self.lim_angulo_4_positivo = None
266     self.lim_angulo_5_positivo = None
267
268     self.lim_angulo_1_negativo = None
269     self.lim_angulo_2_negativo = None
270     self.lim_angulo_3_negativo = None
271     self.lim_angulo_4_negativo = None
272     self.lim_angulo_5_negativo = None
273
274     self.lim_modulo_negativo_1 = None
275     self.lim_modulo_negativo_2 = None
276     self.lim_modulo_negativo_3 = None
277     self.lim_modulo_negativo_4 = None
278     self.lim_modulo_negativo_5 = None
279
280     self.lim_modulo_positivo_1 = None
281     self.lim_modulo_positivo_2 = None
282     self.lim_modulo_positivo_3 = None
283     self.lim_modulo_positivo_4 = None
284     self.lim_modulo_positivo_5 = None
285
286     self.nivel_1_precision = None
287     self.nivel_2_precision = None
288     self.nivel_3_precision = None
289     self.nivel_4_precision = None
290     self.nivel_5_precision = None
291
292     # Definición de las variables de respuesta
293     self.responseDIS3 = None
294     self.response = None
295     # Definición de variables del web service
296     self.prueba = None
297     self.wsdl = None
298     self.client = None
299     self.header_value = None
300
301     # Identificador de nueva tanda
302     self.identificador_nueva_tanda = False
303
304     # SAIO pedir datos dis3
305     def abrir_dis3(self, ov):
306         """M todo que llama a la función ObtenerEspecificacionesDIS3 del
307             web service.
308
309             Esta función nos devuelve un XML con todos los datos referentes a
310             una OV guardados en la base de datos DIS3.
311
312             Parametros:
313                 self — clase ComunicacionWebService
314
315                 ov — Número referente al orden de venta del que se
316                 quieren obtener los datos.
317
318             Atributos:
319
320             """

```

```

316         self.responseDIS3 — respuesta del web service a la
           petición del XML.
317     """
318     self.responseDIS3 = self.client.service.ObtenerEspecificacionesDIS3
           (OV=ov, _soapheaders=self.header_value)
319
320     # SAIO cabecera de conexión
321     def abrir_ws(self, usuario, contraseña):
322         """Método que llama al web service 'https://saio.jakiten.com/
           Ensayos/WebServices/wsPuntosPrecision.asmx?wsdl'.
323
324         Crea una cabecera con usuario y password para poder acceder a ella.
325
326         NOTA: se le añade '?wsdl' al http para poder usar la librería zeep
           para pedir y recibir datos a través de la clase Client.
327
328         Parametros:
329             self — clase ComunicacionWebService
330
331             usuario — Usuario de autenticación.
332
333             password — Contraseña de autenticación.
334
335         Atributos:
336
337             self.wsdl — http del web service de ensayos
338
339             settings — Ajustes de la librería zeep. Recomendado:
                 force_https=False, raw_response=True
340
341             self.client — clase Client
342
343             self.header_value — cabecera de autenticación con Usuario y
                 Contraseña. FORMATO: {"CabeceraAutenticacion":{"Usuario":
                 usuario,"Contraseña":contraseña}}
344
345         """
346         self.wsdl = 'https://saio.jakiten.com/Ensayos/WebServices/
           wsPuntosPrecision.asmx?wsdl'
347         self.prueba = 'https://saio.jakiten.com/EnsayosPrueba/WebServices/
           wsPuntosPrecision.asmx?wsdl'
348         settings = Settings(force_https=False, raw_response=True)
349         self.client = Client(self.wsdl, settings=settings)
350         self.header_value = {"CabeceraAutenticacion": {"Usuario": usuario,
           "Contraseña": contraseña}}
351         response = self.client.service.ComprobarAutenticacion(_soapheaders=
           self.header_value)
352         # Para tener una variable de respuesta guardada para su uso en el
           programa principal.
353         # True si usuario y contraseña correctos. Sino False. (boolean)
354         respuesta = b"true" in response.content
355         return respuesta
356
357     # Nueva tanda
358     def crear_nueva_tanda(self, ov, num_aparato, id_ensayo_etapa_fab,
           id_tanda_nueva):
359         """Método que llama a la función
           CrearNuevaTandaYObtenerDatosEnsayo del web service.

```

```

360
361     Esta funci n crea una nueva tanda de ensayo en SAIO y devuelve los
           datos de una OV, un n mero de aparato , un ID de ensayo y un ID
           de nueva tanda guardados en la base de datos SAIO.
362
363     Parametros:
364         self — clase ComunicacionWebService
365
366         ov — N mero referente al orden de venta del que se
           quieren obtener los datos.
367
368         num_aparato — N mero de aparato
369
370         id_ensayo_etapa_fab — Id del tipo de ensayo. Precisi n
           -->3 Descargas parciales --> 25
371
372         id_tanda_nueva — Id de la nueva tanda. Hace referencia a
           la raz n por la cual se crea una nueva tanda de ensayos
           .
373     Atributos:
374         self.response — respuesta del web service a la petici n
           de una nueva tanda y el nuevo XML.
375
376         self.identificador_nueva_tanda — identificador tipo
           boolean para saber si estamos leyendo los datos de una
           nueva tanda.
377     """
378
379     self.identificador_nueva_tanda = True
380     self.response = self.client.service.
           CrearNuevaTandaYObtenerDatosEnsayo(OV=ov , numAparato=num_aparato
           , idEnsayoEtapaFab=id_ensayo_etapa_fab ,
381     idTandaNueva=id_tanda_nueva , _soapheaders=self.header_value)
382
383     # SAIO obtener datos ensayo
384     def abrir_xml(self , ov , num_aparato , id_ensayo_etapa_fab):
385         """M todo que llama a la funci n ObtenerDatosEnsayo del web
           service.
386
387         Esta funci n nos devuelve un XML con todos los datos referentes a
           una OV, un n mero de aparato y un ID de ensayo guardados en la
           base de datos SAIO.
388
389         Parametros:
390             self — clase ComunicacionWebService
391
392             ov — N mero referente al orden de venta del que se
               quieren obtener los datos.
393
394             num_aparato — N mero de aparato
395
396             id_ensayo_etapa_fab — Id del tipo de ensayo. Precisi n
               -->3 Descargas parciales --> 25
397     Atributos:
398         self.response — respuesta del web service a la petici n
           del XML.
399

```

```

400         self.identificador_nueva_tanda — identificador tipo
401         boolean para saber si estamos leyendo los datos de
402         una nueva tanda.
403     """
404     # Refrescar identificador de nueva tanda
405     self.identificador_nueva_tanda = False
406
407     self.response = self.client.service.ObtenerDatosEnsayo(OV=ov,
408         numAparato=num_aparato, idEnsayoEtapaFab=id_ensayo_etapa_fab,
409         _soapheaders=self.header_value)
410     self.response.content
411
412     # SAIO guardar datos ensayo precision
413     def guardar_xml(self, error_modulo1, error_modulo2, error_modulo3,
414         error_modulo4, error_modulo5, error_angulo1, error_angulo2,
415         error_angulo3, error_angulo4, error_angulo5):
416         """Metodo que prepara la respuesta de ensayo y se la manda al web
417         service mediante la funci n GuardarResultadoPuntosPrecision_STR
418         .
419
420         La funci n GuardarResultadoPuntosPrecision_STR recibe un str con formato
421         xml con los errores en modulo y angulo guardados y los guarda en
422         la base de datos SAIO.
423
424         Parametros:
425         self — clase ComunicacionWebService
426
427         error_modulo1 — error en modulo del ensayo en el primer
428         punto
429
430         error_modulo2 — error en modulo del ensayo en el segundo
431         punto
432
433         error_modulo3 — error en modulo del ensayo en el tercer
434         punto
435
436         error_modulo4 — error en modulo del ensayo en el cuarto
437         punto
438
439         error_modulo5 — error en modulo del ensayo en el quinto
440         punto
441
442         error_angulo1 — error en angulo del ensayo en el primer
443         punto
444
445         error_angulo2 — error en angulo del ensayo en el segundo
446         punto
447
448         error_angulo3 — error en angulo del ensayo en el tercer
449         punto
450
451         error_angulo4 — error en angulo del ensayo en el cuarto
452         punto
453
454         error_angulo5 — error en angulo del ensayo en el quinto
455         punto
456
457         Atributos:

```

```

439         respuesta — los datos que se le van al web service.
440
441     Parametros de salida:
442         response2 — respuesta del web service. Nos devolvera True
            si los datos han sido guardados correctamente y False si
            ha habido un error (e.j. los errores ya estaban
            guardados en SAIO y hemos sobrescrito los mismos
            valores)
443     """
444     respuesta = self.xml_respuesta(error_modulo1, error_modulo2,
            error_modulo3, error_modulo4, error_modulo5, error_angulo1,
            error_angulo2, error_angulo3, error_angulo4, error_angulo5)
445     respuesta.pop('@xmlns', 'https://saio.jakiten.com/ensayosPrueba/
            webservices/')
446     respuesta = xmldict.unparse(respuesta).removeprefix('<?xml
            version="1.0" encoding="utf-8"?>').strip()
447
448     response2 = self.client.service.GuardarResultadoEnsayo_STR(paramXML
            =respuesta, idLabo=47, _soapheaders=self.header_value)
449     return response2
450
451     # SAIO guardar datos ensayo ddpp
452     def guardar_xml_ddpp(self, valor_ddpp_1, valor_ddpp_2, valor_ddpp_3,
            valor_ddpp_4, valor_ddpp_5):
453         """Metodo que prepara la respuesta de ensayo y se la manda al web
            service mediante la funci n GuardarResultadoEnsayo_STR
454
455         La funci n GuardarResultadoEnsayo_STR recibe un str con formato
            xml con los valores en descargas parciales guardados y los
            guarda en la base de datos SAIO.
456
457         Parametros:
458             self — clase ComunicacionWebService
459
460             valor_ddpp_1 — valor de la descarga parcial en pC del
                primer punto
461
462             valor_ddpp_2 — valor de la descarga parcial en pC del
                primer punto
463
464             valor_ddpp_3 — valor de la descarga parcial en pC del
                primer punto
465
466             valor_ddpp_4 — valor de la descarga parcial en pC del
                primer punto
467
468             valor_ddpp_5 — valor de la descarga parcial en pC del
                primer punto
469
470         Atributos:
471             respuesta — los datos que se le dan al web service.
472
473         Parametros de salida:
474             response2 — respuesta del web service. Nos devolvera True
                si los datos han sido guardados correctamente y False si
                ha habido un error (e.j. los errores ya estaban
                guardados en SAIO y hemos sobrescrito los mismos
                valores)

```

```

475     """
476
477     respuesta = self.xml_respuesta_ddpp(valor_ddpp_1, valor_ddpp_2,
478         valor_ddpp_3, valor_ddpp_4, valor_ddpp_5)
479     respuesta.pop('@xmlns', 'https://saio.jakiten.com/ensayosPrueba/
480         webservices/')
481     respuesta = xmldict.unparse(respuesta).removeprefix('<?xml
482         version="1.0" encoding="utf-8"?>').strip()
483
484     response2 = self.client.service.GuardarResultadoEnsayo_STR(paramXML
485         =respuesta, idLabo=47, _soapheaders=self.header_value)
486
487     return response2
488
489 # SAIO generar el xml con resultados ensayo precision
490 def xml_respuesta(self, error_modulo1, error_modulo2, error_modulo3,
491     error_modulo4, error_modulo5, error_angulo1, error_angulo2,
492     error_angulo3, error_angulo4, error_angulo5):
493     """Metodo que graba en el XML recibido los valores de los errores
494     de ensayo.
495
496     Parametros:
497         self — clase ComunicacionWebService
498         error_modulo1 — error en modulo del ensayo en el primer
499             punto
500         error_modulo2 — error en modulo del ensayo en el segundo
501             punto
502         error_modulo3 — error en modulo del ensayo en el tercer
503             punto
504         error_modulo4 — error en modulo del ensayo en el cuarto
505             punto
506         error_modulo5 — error en modulo del ensayo en el quinto
507             punto
508         error_angulo1 — error en angulo del ensayo en el primer
509             punto
510         error_angulo2 — error en angulo del ensayo en el segundo
511             punto
512         error_angulo3 — error en angulo del ensayo en el tercer
513             punto
514         error_angulo4 — error en angulo del ensayo en el cuarto
515             punto
516         error_angulo5 — error en angulo del ensayo en el quinto
517             punto
518
519     Atributos:
520         doc2 — XML recibido convertido a tipo diccionario

```



```

515         datos — subdiccionario donde se encuentran las variables
           de error de angulo y modulo
516
517     Parametros de salida:
518         response: subdiccionario con cabecera '
           ObtenerPuntosPrecisionResult'. Necesario como respuesta
           para el m todo guardar_xml
519     """
520     doc2 = xmldict.parse(self.response.content)
521     datos = doc2['soap:Envelope']['soap:Body']['
           ObtenerDatosEnsayoResponse']['ObtenerDatosEnsayoResult']['diffgr
           :diffgram']['DocumentElement']['
           ENSAYO_VARIABLEES_MULTIPLES_POR_MARCAJE']
522     datos[0]['ERROR_MODULO'] = error_modulo1
523     datos[0]['ERROR_ANGULO'] = error_angulo1
524     datos[1]['ERROR_MODULO'] = error_modulo2
525     datos[1]['ERROR_ANGULO'] = error_angulo2
526     datos[2]['ERROR_MODULO'] = error_modulo3
527     datos[2]['ERROR_ANGULO'] = error_angulo3
528     datos[3]['ERROR_MODULO'] = error_modulo4
529     datos[3]['ERROR_ANGULO'] = error_angulo4
530     datos[4]['ERROR_MODULO'] = error_modulo5
531     datos[4]['ERROR_ANGULO'] = error_angulo5
532     response = doc2['soap:Envelope']['soap:Body']['
           ObtenerDatosEnsayoResponse']
533     return response
534
535     # SAIO generar el xml con resultados ensayo ddpp
536     def xml_respuesta_ddpp(self, valor_ddp_1, valor_ddp_2, valor_ddp_3,
537         valor_ddp_4, valor_ddp_5):
538         """Metodo que graba en el XML recibido los valores de las descargas
           parciales.
539
540         Parametros:
541             self — clase ComunicacionWebService
542
543             valor_ddpp_1 — valor de la descarga parcial en pC del
           primer punto
544
545             valor_ddpp_2 — valor de la descarga parcial en pC del
           primer punto
546
547             valor_ddpp_3 — valor de la descarga parcial en pC del
           primer punto
548
549             valor_ddpp_4 — valor de la descarga parcial en pC del
           primer punto
550
551             valor_ddpp_5 — valor de la descarga parcial en pC del
           primer punto
551
552         Atributos:
553             doc2 — XML recibido convertido a tipo diccionario
554
555             datos — subdiccionario donde se encuentran las variables
           de error de angulo y modulo
556
557     Parametros de salida:

```

```

558         response: subdiccionario con cabecera '
                    ObtenerPuntosPrecisionResult'. Necesario como respuesta
                    para el m todo guardar_xml
559     """
560     doc2 = xmldict.parse(self.response.content)
561     datos = doc2['soap:Envelope']['soap:Body']['
                    ObtenerDatosEnsayoResponse']['ObtenerDatosEnsayoResult']['diffgr
                    :diffgram']['DocumentElement']['
                    ENSAYO_VARIABLES_MULTIPLES_POR_APARATO']
562     datos[0]['VALOR_DP'] = valor_ddp_1
563     datos[1]['VALOR_DP'] = valor_ddp_2
564     datos[2]['VALOR_DP'] = valor_ddp_3
565     datos[3]['VALOR_DP'] = valor_ddp_4
566     datos[4]['VALOR_DP'] = valor_ddp_5
567
568     response = doc2['soap:Envelope']['soap:Body']['
                    ObtenerDatosEnsayoResponse']
569     return response
570
571 def leer_xml_ensayo(self, ensayo):
572     """Funci n que lee el fichero xml de SAIO lo convierte a
                    diccionario y guarda los datos en el atributo que le corresponde
                    .
573
574     Parametros:
575         self — clase ComunicacionWebService
576
577     Atributos:
578         doc2 — XML recibido convertido a tipo diccionario
579
580         datos — subdiccionario donde se encuentran las variables
                    de error de angulo y modulo
581
582     """
583     doc = xmldict.parse(self.response.content)
584
585     # Inicializar datos con el valor completo del xml
586     datos = doc
587     if not self.identificador_nueva_tanda:
588         if ensayo == ENSAYO_PRECISION:
589             # Asignar a datos la estructura del xml con los datos del
                    ensayo de precisi n
590             datos = doc['soap:Envelope']['soap:Body']['
                    ObtenerDatosEnsayoResponse']['ObtenerDatosEnsayoResult'
                    ]['diffgr:diffgram']['DocumentElement']['
                    ENSAYO_VARIABLES_MULTIPLES_POR_MARCAJE']
591         elif ensayo == ENSAYO_DDPP:
592             # Asignar a datos la estructura del xml con los datos del
                    ensayo de descargas
593             datos = doc['soap:Envelope']['soap:Body']['
                    ObtenerDatosEnsayoResponse']['ObtenerDatosEnsayoResult'
                    ]['diffgr:diffgram']['DocumentElement']['
                    ENSAYO_VARIABLES_MULTIPLES_POR_APARATO']
594         elif self.identificador_nueva_tanda:
595             if ensayo == ENSAYO_PRECISION:
596                 # Asignar a datos la estructura del xml con los datos del
                    ensayo de precisi n

```

```

597         datos = doc[ 'soap:Envelope' ][ 'soap:Body' ][ '
        CrearNuevaTandaYObtenerDatosEnsayoResponse' ][ '
598         CrearNuevaTandaYObtenerDatosEnsayoResult' ][
        'diffgr:diffgram' ][ 'DocumentElement' ][ '
        ENSAYO_VARIABLES_MULTIPLES_POR_MARCAJE' ]
599     elif ensayo == ENSAYO_DDPP:
600         # Asignar a datos la estructura del xml con los datos del
        ensayo de descargas
601         datos = doc[ 'soap:Envelope' ][ 'soap:Body' ][ '
        CrearNuevaTandaYObtenerDatosEnsayoResponse' ][ '
        CrearNuevaTandaYObtenerDatosEnsayoResult' ][ 'diffgr:
        diffgram' ][ 'DocumentElement' ][ '
        ENSAYO_VARIABLES_MULTIPLES_POR_APARATO' ]

602
603     try:
604         self.ID_DATOS_ENSAYO = datos[0][ 'ID_DATOS_ENSAYO' ]
605         self.OV = datos[0][ 'OV' ]
606         self.NUM_APARATO = datos[0][ 'NUM_APARATO' ]
607         self.ENSAYO_ETAPA_FAB = datos[0][ 'ENSAYO_ETAPA_FAB' ]
608         self.DESCRIPCION_TANDA = datos[0][ 'DESCRIPCION_TANDA' ]
609         self.NUM_TANDA = datos[0][ 'NUM_TANDA' ]

610
611         if ensayo == ENSAYO_DDPP:
612             if 'VALOR_DP' in datos[0]:
613                 self.valor_ddpp_1 = datos[0][ "VALOR_DP" ]
614                 self.valor_ddpp_2 = datos[1][ "VALOR_DP" ]
615                 self.valor_ddpp_3 = datos[2][ "VALOR_DP" ]
616                 self.valor_ddpp_4 = datos[3][ "VALOR_DP" ]
617                 self.valor_ddpp_5 = datos[4][ "VALOR_DP" ]
618             else:
619                 self.valor_ddpp_1 = None
620                 self.valor_ddpp_2 = None
621                 self.valor_ddpp_3 = None
622                 self.valor_ddpp_4 = None
623                 self.valor_ddpp_5 = None
624         elif ensayo == ENSAYO_PRECISION:
625             self.errorModulo1 = datos[0][ 'ERROR_MODULO' ]
626             self.errorModulo2 = datos[1][ 'ERROR_MODULO' ]
627             self.errorModulo3 = datos[2][ 'ERROR_MODULO' ]
628             self.errorModulo4 = datos[3][ 'ERROR_MODULO' ]
629             self.errorModulo5 = datos[4][ 'ERROR_MODULO' ]
630             self.errorAngulo1 = datos[0][ 'ERROR_ANGULO' ]
631             self.errorAngulo2 = datos[1][ 'ERROR_ANGULO' ]
632             self.errorAngulo3 = datos[2][ 'ERROR_ANGULO' ]
633             self.errorAngulo4 = datos[3][ 'ERROR_ANGULO' ]
634             self.errorAngulo5 = datos[4][ 'ERROR_ANGULO' ]
635     except:
636         print("Error leyendo xml")
637
638     # devuelve valor de variable del dis3
639     @staticmethod
640     def asignar_valor_var_dis3(dic, clave):
641         """Funci n que devuelve el valor de la variable asignada en el
        dis3.

642
643     Par metros:
644         dic — diccionario que contiene todas las variables del dis3
        con sus valores.

```

```

645         clave — nombre de la variable de la que queremos el valor
646
647     """
648     if clave in dic.keys():
649         return dic[clave]
650
651 def leer_dis3(self):
652     """Funci n que lee el fichero xml de DIS3 lo convierte a
653     diccionario y guarda los datos en el atributo que le corresponde
654     .
655     Parametros:
656         self — clase ComunicacionWebService
657
658     Atributos:
659         doc2 — XML recibido convertido a tipo diccionario
660
661         datos — subdiccionario donde se encuentran las variables
662         de error de angulo y modulo
663
664     """
665     doc = xmldict.parse(self.responseDIS3.content)
666     datos = \doc['soap:Envelope']['soap:Body']['
667         ObtenerEspecificacionesDIS3Response']['
668         ObtenerEspecificacionesDIS3Result']['diffgr:diffgram']['
669         DocumentElement']['ESPECIFICACIONES_TODAS']
670     # Generar diccionario con los valores de las variables del dis3
671     dis3_vars = {}
672     for i in range(len(datos)):
673         try:
674             dis3_vars[datos[i]['IDENTIFICADOR']] = datos[i]['VALOR']
675         except:
676             pass
677     # Cantidad de aparatos: necesario para hacer l gicas de
678     comprobaci n de la interfaz
679     self.cantidad_aparatos = self.asignar_valor_var_dis3(dis3_vars, '
680     cantidad_aparatos')
681
682     # Informaci n general de la interfaz
683     self.carga_capac = self.asignar_valor_var_dis3(dis3_vars, '
684     carga_capac')
685     self.carga_resist = self.asignar_valor_var_dis3(dis3_vars, '
686     carga_resist')
687     self.CLASE = self.asignar_valor_var_dis3(dis3_vars, 'CLASE')
688     self.factor_tension = self.asignar_valor_var_dis3(dis3_vars, '
689     factor_tension')
690     self.modelo = self.asignar_valor_var_dis3(dis3_vars, 'modelo')
691     self.nombre_cliente = self.asignar_valor_var_dis3(dis3_vars, '
692     nombre_cliente')
693     self.norma = self.asignar_valor_var_dis3(dis3_vars, 'norma')
694     self.orden_fab = self.asignar_valor_var_dis3(dis3_vars, 'orden_fab'
695     )
696     self.OV_IN_OUT = self.asignar_valor_var_dis3(dis3_vars, 'OV_IN_OUT'
697     )
698     self.temp_max = self.asignar_valor_var_dis3(dis3_vars, 'temp_max')
699     self.temp_min = self.asignar_valor_var_dis3(dis3_vars, 'temp_min')

```

```

689 # Configuraci n ensayo precisi n. Valores del patron??
690 self.num_t_1_sec = self.asignar_valor_var_dis3(dis3_vars ,
        num_t_1_sec')
691 self.num_t_prim = self.asignar_valor_var_dis3(dis3_vars ,
        num_t_prim')
692 self.frecuencia = self.asignar_valor_var_dis3(dis3_vars ,
        frecuencia')
693
694 # DDPP
695 self.lim_1_ddpp = self.asignar_valor_var_dis3(dis3_vars ,
        lim_1_ddpp')
696 self.lim_2_ddpp = self.asignar_valor_var_dis3(dis3_vars ,
        lim_2_ddpp')
697 self.lim_3_ddpp = self.asignar_valor_var_dis3(dis3_vars ,
        lim_3_ddpp')
698 self.lim_4_ddpp = self.asignar_valor_var_dis3(dis3_vars ,
        lim_4_ddpp')
699 self.lim_5_ddpp = self.asignar_valor_var_dis3(dis3_vars ,
        lim_5_ddpp')
700
701 self.pto_1_ddpp = self.asignar_valor_var_dis3(dis3_vars ,
        pto_1_ddpp')
702 self.pto_2_ddpp = self.asignar_valor_var_dis3(dis3_vars ,
        pto_2_ddpp')
703 self.pto_3_ddpp = self.asignar_valor_var_dis3(dis3_vars ,
        pto_3_ddpp')
704 self.pto_4_ddpp = self.asignar_valor_var_dis3(dis3_vars ,
        pto_4_ddpp')
705 self.pto_5_ddpp = self.asignar_valor_var_dis3(dis3_vars ,
        pto_5_ddpp')
706
707 # Precisi n
708 self.lim_angulo_1_positivo = self.asignar_valor_var_dis3(dis3_vars ,
        'lim_angulo_1')
709 self.lim_angulo_2_positivo = self.asignar_valor_var_dis3(dis3_vars ,
        'lim_angulo_2')
710 self.lim_angulo_3_positivo = self.asignar_valor_var_dis3(dis3_vars ,
        'lim_angulo_3')
711 self.lim_angulo_4_positivo = self.asignar_valor_var_dis3(dis3_vars ,
        'lim_angulo_4')
712 self.lim_angulo_5_positivo = self.asignar_valor_var_dis3(dis3_vars ,
        'lim_angulo_5')
713
714 try:
715     self.lim_angulo_1_negativo = "-" + self.lim_angulo_1_positivo
716     self.lim_angulo_2_negativo = "-" + self.lim_angulo_2_positivo
717     self.lim_angulo_3_negativo = "-" + self.lim_angulo_3_positivo
718     self.lim_angulo_4_negativo = "-" + self.lim_angulo_4_positivo
719     self.lim_angulo_5_negativo = "-" + self.lim_angulo_5_positivo
720 except:
721     pass
722
723 self.lim_modulo_negativo_1 = self.asignar_valor_var_dis3(dis3_vars ,
        'lim_modulo_negativo_1')
724 self.lim_modulo_negativo_2 = self.asignar_valor_var_dis3(dis3_vars ,
        'lim_modulo_negativo_2')
725 self.lim_modulo_negativo_3 = self.asignar_valor_var_dis3(dis3_vars ,
        'lim_modulo_negativo_3')

```

```
726 self.lim_modulo_negativo_4 = self.asignar_valor_var_dis3(dis3_vars ,  
727     'lim_modulo_negativo_4')  
728 self.lim_modulo_negativo_5 = self.asignar_valor_var_dis3(dis3_vars ,  
729     'lim_modulo_negativo_5')  
730 self.lim_modulo_positivo_1 = self.asignar_valor_var_dis3(dis3_vars ,  
731     'lim_modulo_positivo_1')  
732 self.lim_modulo_positivo_2 = self.asignar_valor_var_dis3(dis3_vars ,  
733     'lim_modulo_positivo_2')  
734 self.lim_modulo_positivo_3 = self.asignar_valor_var_dis3(dis3_vars ,  
735     'lim_modulo_positivo_3')  
736 self.lim_modulo_positivo_4 = self.asignar_valor_var_dis3(dis3_vars ,  
737     'lim_modulo_positivo_4')  
738 self.lim_modulo_positivo_5 = self.asignar_valor_var_dis3(dis3_vars ,  
739     'lim_modulo_positivo_5')  
  
734 self.nivel_1_precision = self.asignar_valor_var_dis3(dis3_vars , '  
735     nivel_1_precision')  
736 self.nivel_2_precision = self.asignar_valor_var_dis3(dis3_vars , '  
737     nivel_2_precision')  
738 self.nivel_3_precision = self.asignar_valor_var_dis3(dis3_vars , '  
739     nivel_3_precision')  
740 self.nivel_4_precision = self.asignar_valor_var_dis3(dis3_vars , '  
741     nivel_4_precision')  
742 self.nivel_5_precision = self.asignar_valor_var_dis3(dis3_vars , '  
743     nivel_5_precision')
```

## D. MaquinaEstados Klasea

```

1 import logging
2 import enum
3
4
5 class MaquinaEstados(object):
6     """Clase que representa la maquina de estados del proyecto
7         autolabo_sensores.
8
9         Consta de 10 estados distintos y su funci n es controlar el flujo de
10        actividad del programa principal pasando de un estado a otro de
11        forma controlada. En cada estado se ejecutan las funciones
12        correspondientes a esta.
13
14        """
15
16     class Estados(enum.Enum):
17         """Clase anidada de tipo Enum.
18
19         Define los diferentes estados que puede tomar la maquina de
20         estados.
21
22         """
23         P0_LOG_IN = 0
24         P1_SELECCION_TIPO_ENSAYOS = 1
25         P2_COMENZAR_ENSAYO = 2
26         P3_FIN_ENSAYO = 3
27         P4_PARAR_ENSAYO = 4
28         P5_GRABAR_DATOS = 5
29         P6_DESCARTAR_DATOS = 6
30         P7_ESTADO_ESPERA = 7
31         P8_CONFIGURAR_ZERA = 8
32         P9_GRABAR_RESULTADO_INTERFAZ = 9
33
34     def __init__(self, ventana):
35         """
36         Constructor donde se inicializan las variables a usar dentro la
37         maquina de estados.
38
39         Incluye la definici n del estado inicial que debe tomar la
40         maquina de estados.
41
42         Par metros:
43         self — clase MaquinaEstados
44
45         ventana — interfaz del programa principal tipo .ui
46
47         Atributos:
48         self.ventana — referencia al programa principal
49
50         self.logger — atributo tipo Logger para el control y
51         obtenci n de informaci n y errores del programa
52
53         self.estado — estado actual en el que se encuentra la maquina
54         de estados
55
56         self.lectura — atributo de control para saber en qu lectura
57         se encuentra el programa principal

```

```

47
48     """
49     # Guardar referencia al programa principal
50     self.ventana = ventana
51
52     # Definir registro
53     self.logger = logging.getLogger(__name__)
54
55     # Inicializar variables
56     self.estado = MaquinaEstados.Estados.P0_LOG_IN
57     self.lectura = None
58
59     def cambiar_estado(self, nuevo_estado):
60         """
61         Funci n dedicada a realizar la transici n entre estado y estado.
62         Deja un registro de logging cuando estas ocurren.
63
64         Par metros:
65         self — clase M quinaEstados
66
67         nuevo_estado — el nuevo estado al que se quiere cambiar
68
69         Atributos:
70         self.estado — estado actual en el que se encuentra la
71         m quina de estados
72
73         """
74         # Registro en el logging
75         self.logger.info('Cambiando estado ' + str(self.estado) + ' -> ' +
76                         str(nuevo_estado))
77
78         # Cambio del estado actual
79         self.estado = nuevo_estado
80
81     def run(self):
82         """
83         Funci n principal de la m quina de estados que permite controlar
84         el flujo del programa principal.
85
86         Comprueba el estado en el que se encuentra el programa y ejecuta
87         las funciones necesarias en cada caso.
88
89         En caso de haber alg n error lo recoge y lo ense a en una ventana
90         de interfaz.
91
92         Par metros:
93         self — clase MaquinaEstados
94         """
95         try:
96             if self.estado == MaquinaEstados.Estados.P0_LOG_IN:
97                 self.ventana.log_in()
98
99             elif self.estado == MaquinaEstados.Estados.
100                 P1_SELECCION_TIPO_ENSAYOS:
101                 self.activar_seleccion_tipo_ensayos()
102                 self.cambiar_estado(MaquinaEstados.Estados.P7_ESTADO_ESPERA
103                                     )

```



```

97     elif self.estado == MaquinaEstados.Estados.P2_COMENZAR_ENSAYO:
98         self.cambiar_estado(MaquinaEstados.Estados.P7_ESTADO_ESPERA
99             )
100         self.ventana.comenzar_ensayo()
101         self.lectura = 0
102
103     elif self.estado == MaquinaEstados.Estados.P3_FIN_ENSAYO:
104         self.ventana.fin_ensayo()
105         self.activar_grabar_descartar_datos()
106
107     elif self.estado == MaquinaEstados.Estados.P4_PARAR_ENSAYO:
108         self.ventana.parar_ensayo()
109         self.activar_grabar_descartar_datos()
110
111     elif self.estado == MaquinaEstados.Estados.P5_GRABAR_DATOS:
112         self.ventana.grabar_datos_ensayo()
113         self.cambiar_estado(MaquinaEstados.Estados.
114             P1_SELECCION_TIPO_ENSAYOS)
115
116     elif self.estado == MaquinaEstados.Estados.P6_DESCARTAR_DATOS:
117         self.ventana.descartar()
118         self.cambiar_estado(MaquinaEstados.Estados.
119             P1_SELECCION_TIPO_ENSAYOS)
120
121     elif self.estado == MaquinaEstados.Estados.P7_ESTADO_ESPERA:
122         pass
123
124     elif self.estado == MaquinaEstados.Estados.P8_CONFIGURAR_ZERA:
125         self.ventana.configurar_zera()
126         self.cambiar_estado(MaquinaEstados.Estados.
127             P9_GRABAR_RESULTADO_INTERFAZ)
128
129     elif self.estado == MaquinaEstados.Estados.
130         P9_GRABAR_RESULTADO_INTERFAZ:
131         self.ventana.grabar_resultados_precision(self.lectura)
132         self.lectura += 1
133
134     except Exception as e:
135         self.ventana.mostrar_mensaje_error(e.__str__(), e.__str__())
136         self.logger.exception(e)
137
138 def reset(self):
139     """
140     Funci n que resetea la m quina de estados al estado inicial.
141
142     Estado inicial: P0_LOG_IN
143
144     Par metros:
145     self — clase MaquinaEstados
146     """
147     self.cambiar_estado(MaquinaEstados.Estados.P0_LOG_IN)
148
149 def activar_seleccion_tipo_ensayos(self):
150     """
151     Funci n auxiliar del estado P1_SELECCION_TIPO_ENSAYOS
152
153     Activa y desactiva los botones de la interfaz del programa
154     principal referentes a este estado.Para ello llama

```

```

149     a funciones del programa principal.
150
151     Par metros:
152         self — clase MaquinaEstados
153
154     Atributos:
155         ventana — programa principal
156
157     """
158     # Botones
159     self.ventana.activar_boton_cambiar_usuario()
160     self.ventana.activar_boton_ddpp()
161     self.ventana.activar_boton_prec()
162
163     self.ventana.desactivar_boton_grabar()
164     self.ventana.desactivar_boton_descartar()
165     self.ventana.desactivar_boton_inicio()
166     self.ventana.desactivar_boton_parar()
167
168     def activar_grabar_descartar_datos(self):
169         """
170         Funci n auxiliar de los estados P5_GRABAR_DATOS y
171         P6_DESCARTAR_DATOS
172
173         Activa y desactiva los botones de la interfaz del programa
174         principal referentes a este estado. Para ello llama
175         a funciones del programa principal.
176
177         Par metros:
178             self — clase MaquinaEstados
179
180         Atributos:
181             ventana — programa principal
182
183         """
184         # Habilitar botones
185         self.ventana.activar_boton_grabar()
186         self.ventana.activar_boton_descartar()
187
188         # Deshabilitar botones
189         self.ventana.desactivar_boton_parar()

```

## E. App Klasea

```

1 import logging
2 import sys
3
4 from PyQt5 import uic, QtWidgets, QtCore
5 from PyQt5.QtGui import QPixmap
6
7 from zera import PuenteZera
8 from comunicacion_web_service import ComunicacionWebService
9 from hipotronics import Hipotronics
10 from estados import MaquinaEstados
11 import venatan_log_in
12
13 # VARIABLES GLOBALES A futuro deben tomar los valores de un fichero de
14   # configuraci n
15 ENSAYO_DDPP = 25 # Identificacor del ensayo DDPP
16 ENSAYO_PRECISION = 3 # Identificador del ensayo PRECISION
17 HORQUILLA = "5" # Valor de la horquilla aceptable en tensi n de los
18   # puntos de medida de precisi n. Valor habitual 5.
19 PATRON_PRIMARIO = "70000" # tensi n primaria del patron. El valor
20   # habitual es 70000.
21 PATRON_SECUNDARIO = "110" # tensi n secundaria del patron. El valor
22   # habitual es 2.
23
24 # En caso de querer crear una nueva tanda se crean con la siguiente
25   # descripci n. Cada uno tiene su ID (c digo asociado)
26 OPCIONES_NUEVA_TANDA = ["", "Probado despu s de rehacer", "Probado
27   # despu s de reparaci n", "Repetici n de ensayo"]
28 ID_PROBADO_REHACER = 5
29 ID_PROBADO_REPARACION = 9
30 ID_REPETICION = 65
31
32 # En un QDialog de la libreria PyQt5 al clicar un bot n la funci n
33   # .exec() devuelve un valor int diferente para cada bot n clicado.
34 CLICAR_YES = 16384 # Valor que devuelve al clicar el bot n Yes (Standard
35   # Button)
36 CLICAR_NO = 65536 # Valor que devuelve al clicar el bot n No (Standard
37   # Button)
38
39 qtCreatorFile = "ui/main.ui" # Nombre del archivo de la interfaz .ui (
40   # PyQt5—> Qt Designer)f
41 Ui_MainWindow, QtBaseClass = uic.loadUiType(qtCreatorFile)
42
43 logging.basicConfig(level=logging.INFO)
44
45 class App(QtWidgets.QMainWindow, Ui_MainWindow):
46     """Clase de la interfaz del proyecto creado con PyQt5.
47
48     La interfaz posibilita la comunicaci n con dispositivos y bases de
49     datos y su flujo se controla mediante una m quina de estados.
50
51     General:
52     1.—Se piden usuario y contrase a (necesarios para el acceso a
53     las bases de datos)
54

```

45 2.—Se escriben la OV y el número de aparato.  
46  
47 4.—Se elige entre los ensayos Precisión o DDPP (descargas  
48 parciales)  
49 5.—Se cargan los datos correspondientes al ensayo desde la base  
50 de datos.  
51 6.— Al pulsar HV On empieza el proceso : activar HIPOTRONICS y  
52 grabar en pantalla resultados del ZERA en los puntos  
53 correspondientes en el caso de precisión. En caso de  
54 descargas parciales los resultados se escriben de forma  
55 manual.  
56 7.— Al pulsar HV Off se para el proceso.  
57 8.— Al pulsar GRABAR se graban los resultados en SAIO.  
58 9.— Al pulsar DESCARTAR se limpia toda la pantalla.

59 *Comprobaciones:*  
60 1. Se comprueba usuario y contraseña  
61  
62 2. Se verifica que la OF tenga 11 dígitos  
63  
64 3. Se comprueba si la OF ya está ensayada y se da opción a una  
65 nueva tanda

66 *Control de errores:*  
67 Error Conexión ZERA — No se ha podido establecer conexión  
68 TCP/IP con el dispositivo.  
69 Error Conexión HIPOTRONICS — No se ha podido establecer  
70 conexión TCP/IP con el dispositivo.  
71 Error Introducci3n de Datos — Usuario, contraseña o OF  
72 incorrectos  
73 Error Guardar — No se han podido guardar los datos en la base  
74 de datos.  
75 Error Lectura de Datos — No se han podido leer los datos  
76 correctamente de las bases de datos

77 ""  
78  
79 **def** `__init__`(self):  
80 *"""Funci3n por defecto de inicializaci3n de objeto dentro de la  
81 clase.*

82 *Inicializaciones:*  
83 1. Logger  
84  
85 2. Interfaz (main.ui)  
86  
87 3. Clases: ComunicacionWebService, PuenteZera, Hipotronics  
88

89           4. *M quina de estados. Se ejecuta la funci n run de la*  
               *m quina de estados cada 10 milisegundos haciendo uso de un*  
               *Timer de PyQt5.*

90

91           *Acciones secundarias:*

92            – *Crear grupos de controles de la interfaz*

93            – *Conectar botones a cada acci n*

94            – *Aadir logo*

95            – *Inicializar en la interfaz las casillas de datos de entrada*

96            – *Escribir en el logger el lanzamiento del programa*

97

98           *Excepciones:*

99            *En caso de ocurrir cualquier excepci n se ense ar en*  
               *pantalla*

100

101           *Par metros:*

102            *self* — *Clase App*

103

104           *Atributos:*

105            *Generales:*

106                *self.logger* — *Logger*

107

108                *self.xml* — *clase ComunicacionWebService*

109

110                *self.pz* — *clase PuenteZera*

111

112                *self.h* — *clase Hipotronics*

113

114                *self.maquina\_estados* — *clase MaquinaEstados*

115

116            *Log in:*

117                *self.usuario* — *usuario para el log in*

118

119                *self.password* — *contrase a para el log in*

120

121            *Selecci n tipo de estado:*

122                *self.ensayo* — *tipo de ensayo seleccionado que puede ser*  
                   *precisi n o descargas parciales*

123

124                *self.ov* — *Orden de venta: primeros 8 d gitos de la OF*

125

126                *self.num\_aps* — *lista de n mero de aparatos: ltimos 3*  
                   *d gitos de la OF*

127

128            *Grupos de controles de la interfaz:*

129                *self.controles\_datos\_entrada* — *lista que hace referencia a*  
                   *los datos de entrada*

130

131                *self.controles\_hipotronics* — *lista que hace referencia a*  
                   *los datos del Hipotronics*

132

133                *self.controles\_lecturas\_zera* — *lista que hace referencia a*  
                   *los datos de las lecturas del puente Zera*

134

135                *self.controles\_configuracion\_zera* — *lista que hace*  
                   *referencia a los datos de la configuraci n del puente*  
                   *Zera*

136

```

137         self.controles_info_general — lista que hace referencia a
138             la informaci n general de la OF
139         self.controles_lecturas_ddpp — lista que hace referencia a
140             los datos de las medidas de descargas parciales
141         self.controles_lecturas_precision — lista que hace
142             referencia a los datos de las medidas de precision
143         self.controles_limite_negativo_angulo — lista que hace
144             referencia a los limites negativos del error en
145             ngulo para ensayo de precisi n
146         self.controles_limite_negativo_modulo — lista que hace
147             referencia a los limites negativos del error en
148             m dulo para ensayo de precisi n
149         self.controles_limite_positivo_angulo — lista que hace
150             referencia a los limites positivos del error en
151             ngulo para ensayo de precisi n
152         self.controles_limite_positivo_modulo — lista que hace
153             referencia a los limites positivos del error en
154             m dulo para ensayo de precisi n
155         self.controles_limites_ddpp — lista que hace referencia a
156             los limites del error en descargas parciales
157         self.controles_valores_error_modulo — lista que hace
158             referencia a los errores en m dulo para ensayo de
159             precisi n
160         self.controles_valores_error_angulo — lista que hace
161             referencia a los errores en ngulo para ensayo de
162             precisi n
163
164     """
165     try:
166         # Iniciar logger
167         self.logger = logging.getLogger(__name__)
168
169         # Setup control window
170         QtWidgets.QMainWindow.__init__(self)
171         Ui_MainWindow.__init__(self)
172         self.setupUi(self)
173         self.setWindowFlag(QtCore.Qt.WindowMinimizeButtonHint, on=True)
174         self.setWindowFlag(QtCore.Qt.WindowMaximizeButtonHint, on=True)
175
176         # Definir los atributos de las clases a utilizar
177         self.xml = ComunicacionWebService()
178         self.pz = PuenteZera()
179         self.h = Hipotronics()
180
181         # Inicializar m quina de estados
182         self.maquina_estados = MaquinaEstados(ventana=self)
183         self.maquina_estados.reset()
184

```

```

185     # Ejecutar m quina de estados cada 0.01 secs
186     self.timer1 = QtCore.QTimer(self)
187     self.timer1.timeout.connect(self.maquina_estados.run)
188     self.timer1.start(50)
189
190     # Log in
191     # Atributos
192     self.usuario = None
193     self.password = None
194
195     # Selecci n tipo de ensayo
196     # Atributos
197     self.ensayo = None
198     self.ov = None
199     self.num_aps = []
200
201     # Grupos de controles de la interfaz
202     # La idea es agrupar los controles en listas para aplicarles
203     # cambios a la vez
204     self.controles_datos_entrada = []
205     self.controles_hipotronics = []
206     self.controles_lecturas_zera = []
207     self.controles_configuracion_zera = []
208     self.controles_info_general = []
209     self.controles_lecturas_ddpp = []
210     self.controles_lecturas_precision = []
211     self.controles_limite_negativo_angulo = []
212     self.controles_limite_negativo_modulo = []
213     self.controles_limite_positivo_angulo = []
214     self.controles_limite_positivo_modulo = []
215     self.controles_limites_ddpp = []
216     self.controles_valores_error_modulo = []
217     self.controles_valores_error_angulo = []
218
219     self.crear_grupos_controles()
220
221     # Botones
222     self.configurar_botones()
223
224     # Interfaz
225     # Logo
226     self.add_logo()
227     # Datos entrada
228     self.inicializar_datos_entrada()
229
230     # Logger
231     self.logger.info('Lanzado programa principal')
232
233     except Exception as e:
234         self.mostrar_mensaje_error(e.__str__(), e.__str__())
235         self.logger.exception(e)
236
237     # Funciones del flujo principal
238     def log_in(self):
239         """Funci n que recoge los m todos en orden del log in. Estado

```

```

240     1. Pedir usuario y contrase a mediante pantalla de seguridad
        secundaria
241     2. Comprobar usuario y contrase a
242     3. Refrescar la pantalla principal
243
244     Par metros:
245         self — clase App
246
247     """
248     self.obtener_datos_usuario_contrasena()
249     self.comprobar_usuario_y_contrasena()
250
251     # Interfaz pantalla principal
252     self.refrescar_pantalla_log_in()
253
254     def seleccion_tipo_ensayo(self):
255         """Funci n que recoge los m todos en orden de la selecci n del
        tipo de ensayo. Estado: P1_SELECCION_TIPO_ENSAYOS
256
257         1. Comprobar si la OF es correcta. En caso de ser correcta:
258         2. Intentar obtener datos del ensayo del DIS3 y de SAIO. Si
        la lectura de datos es correcta:
259         3. Mostrara informaci n general de OF en pantalla
260         4. Activar bot n descartar
261
262         Par metros:
263             self — clase App
264
265         Devuelve:
266             True — si la OF y la lectura de datos es correcta
267
268             False — en caso contrario
269         """
270         if self.comprobar_ofs():
271             if self.obtener_datos_ensayo():
272                 self.mostrar_info_general()
273                 self.activar_boton_descartar()
274                 return True
275         else:
276             return False
277
278     def comenzar_ensayo(self):
279         """Funci n que recoge los m todos en orden del comienzo del
        ensayo. Estado: P2_COMENZAR_ENSAYO
280
281         1. Configurar botones de la interfaz
282         2. Comenzar conexi n con Hipotronics
283
284         En caso de ensayo de precisi n:
285         3. Intentar conectar con el Zera. En caso de que haya conexi n
        cambiar estado a configurar zera. Estado nuevo:
        P8_CONFIGURAR_ZERA
286
287         Para cualquier otro ensayo:
288         3. Cambiar estado a final de ensayo. Estado nuevo:
        P3_FIN_ENSAYO
289
290         Par metros:

```



```

291         self — clase App
292
293     Atributos:
294         self.ensayo — tipo de ensayo seleccionado que puede ser
                precisi n o descargas parciales
295
296         self.pz — clase PuenteZera
297
298     """
299     # Habilitar boton de parar por emergencia
300     self.activar_boton_parar()
301     # Deshabilitar el resto
302     self.desactivar_boton_ddpp()
303     self.desactivar_boton_inicio()
304     self.desactivar_boton_prec()
305     self.desactivar_boton_descartar()
306     self.desactivar_boton_cambiar_usuario()
307
308     # Comenzar conexi n hipotronics
309     self.comenzar_conexion_hipotronics()
310     # En caso de ensayo de precisi n conectar zera, configurarlo y
                leer los datos
311     if self.ensayo == ENSAYO_PRECISION:
312         self.pz.conectar_zera()
313         if self.pz.connected == 1:
314             # Enviar configuraci n al Zera
315             self.cambiar_estado_configurar_zera()
316         else:
317             self.mostrar_mensaje_error("Error conexi n ZERA", "No se
                ha podido conectar con el ZERA.")
318     else:
319         self.cambiar_estado_fin_ensayo()
320
321     def fin_ensayo(self):
322         """Funci n que recoge los m todos en orden del fin del ensayo.
                Estado: P3_FIN_ENSAYO.
323
324         1. Desconecta Zera
325         2. Desconecta el Hipotronics
326
327         Par metros:
328             self — clase App
329
330         Atributos:
331             self.pz — clase PuenteZera
332
333             self.h — clase Hipotronics
334
335         """
336         # Desconectamos ZERA
337         self.pz.desconectar_zera()
338         # Desconectamos Hipotronics
339         self.h.desconectar_hipotronics()
340
341     def grabar_datos_ensayo(self):
342         """Funci n que recoge los m todos en orden del grabado de datos
                del ensayo. Estado: P5_GRABAR_DATOS
343

```

```

344     1. Mira que tipo de ensayo es (precisi n o descargas parciales)
345     2. Guarda los datos en la base de datos que le corresponde seg n
        el tipo de ensayo
346     3. Configura los botones de la interfaz
347
348     Par metros:
349         self — clase App
350
351     Atributos:
352         self.ensayo — tipo de ensayo seleccionado que puede ser
        precisi n o descargas parciales
353     """
354     if self.ensayo == ENSAYO_DDPP:
355         self.guardar_datos_descargas_parciales()
356     if self.ensayo == ENSAYO_PRECISION:
357         self.guardar_datos_precision()
358
359     # Desactivar botones
360     self.desactivar_boton_grabar()
361     self.desactivar_boton_descartar()
362
363     # Activar botones
364     self.activar_boton_cambiar_usuario()
365     self.activar_boton_prec()
366     self.activar_boton_ddpp()
367
368     def descartar(self):
369         """Funci n que recoge los m todos en orden del descarte de datos
        del ensayo. Estado: P6_DESCARTAR_DATOS
370
371         1. Limpia todos los datos de la interfaz
372         2. Configura los botones
373
374         Par metros:
375             self — clase App
376
377         """
378         self.limpiar_info_general()
379         # controles_hipotronic
380         for c in self.controles_hipotronics:
381             self.texto_a_blanco(c)
382             self.inicializar_datos_entrada()
383             self.limpiar_datos_ddpp()
384             self.limpiar_datos_precision()
385
386         # Desactivar botones
387         self.desactivar_boton_grabar()
388         self.desactivar_boton_descartar()
389
390         # Activar botones
391         self.activar_boton_cambiar_usuario()
392         self.activar_boton_prec()
393         self.activar_boton_ddpp()
394
395     def parar_ensayo(self):
396         """Funci n que recoge los m todos en orden de la acci n de parar
        el ensayo. Estado: P4_PARAR_ENSAYO
397

```

```

398     1. Comprobar si Hipotronics conectado. Si conectado:
399         2. Bajar tensi n a 0
400     3. Desconectar Zera
401     4. Desconectar Hipotronics
402
403     Par metros:
404         self — clase App
405
406     Atributos:
407         self.pz — clase PuenteZera
408
409         self.h — clase Hipotronics
410     """
411     # Si el Hipotronics est conectado hacemos bajar la tensi n de
412     # forma segura
413     if self.h.connected == 1:
414         self.h.set_hv_off()
415     # Desconectamos los socket
416     self.pz.desconectar_zera()
417     self.h.desconectar_hipotronics()
418
419     # Funciones del log in
420     def obtener_datos_usuario_contrasena(self):
421         """M todo para creaci n del pop up dialog de Seguridad.
422
423         De aqu obtenemos el usuario y la contrasena.
424
425         Par metros:
426             self — clase App
427
428         Atributos:
429             input_dialog — tupla de strings que contiene el usuario y la
430             contrase a obtenidas de la pantalla del log in.
431
432             self.usuario — usuario para el log in
433
434             self.password — contrase a para el log in
435         """
436         input_dialog = self.pantalla_log_in()
437         self.usuario = input_dialog[0]
438         self.password = input_dialog[1]
439
440     def comprobar_usuario_y_contrasena(self):
441         """M todo que comprueba el usuario y la contrase a.
442
443         1. Comprobar usuario y contrase a. En caso de no ser correctos
444         alguno de los dos:
445             2. Ense a una ventana con aviso de usuario o contrase a
446             incorrectos
447
448         Si son correctos:
449             2. Cambia el estado a la de selecci n del tipo de ensayo.
450             Nuevo estado: P1_SELECCION_TIPO_ENSAYOS
451
452         Par metros:
453             self — clase App
454
455         Atributos:

```

```

451         self.xml — clase ComunicacionWebService
452
453     Errores:
454         Error de Introducci n de Datos — Usuario o contrase a
455             incorrectos
456         """
457     # Comprobar usuario y contrase a
458     if not self.xml.abrir_ws(self.usuario , self.password):
459         self.show_dialog_error_usuario_contrasena()
460     else:
461         self.cambiar_estado_seleccion_tipo_ensayo()
462
463     # Funciones de selecci n tipo de ensayos
464     def comprobar_ofs(self):
465         """M todo que comprueba la OF.
466
467         1. Recoge la OF de la interfaz
468         2. Comprueba que la OF tenga la largura correcta (11 d gitos). Si
469             no:
470             3. Muestra mensaje de error
471             4. Pone en rojo la casilla de la OF
472             5. Devuelve False
473         Si s :
474             3. Crea una lista con todas las OFs de la casilla (separadas
475                 por /n)
476             4. Comprueba que el n mero de OFs no supera el m ximo (
477                 m ximo 6). Si supera el m ximo:
478                 5. Muestra mensaje de error
479             Si no lo supera:
480                 5. Comprueba el tipo de ensayo
481                 6. Escribe la OV en la interfaz (primeros 8 d gitos de la
482                     OF)
483                 7. Escribe el n mero de aparato de la primera OF en la
484                     interfaz ( ltimos 3 d gitos de la OF)
485                 8. Pone fondo blanco a la casilla de la OF
486                 9. Devuelve True
487
488         Par metros:
489         self — clase App
490
491         Atributos:
492         of — Orden de fabricaci n. Consta de 11 d gitos. Debe ser
493             introducido por el operario.
494
495         lista_ofs — lista compuesta por todas las OFs introducidas por
496             el operario. N mero m ximo de OFs -> 6
497
498         self.ensayo — tipo de ensayo seleccionado que puede ser
499             precisi n o descargas parciales
500
501         self.textEdit_OFS — textEdit de la interfaz main.ui que hace
502             referencia a la casilla de OF.
503         Clase QTextEdit de la librer a Qt5.
504
505         self.textEdit_OV — textEdit de la interfaz main.ui que hace
506             referencia a la casilla de OV.
507         Clase QTextEdit de la librer a Qt5.

```

```

498         self.textEdit_NA — textEdit de la interfaz main.ui que hace
499             referencia a la casilla deL n mero de aparato.
500         Clase QTextEdit de la librer a Qt5.
501
502     Devuelve:
503         False — Si la OF tiene menos de 11 d gitos
504
505         True — Si la OF tiene 11 o m s d gitos
506
507     Errores:
508         Error Introducci n de Datos — OF incorrectos
509         """
510     # empezar con una sola of
511     of = self.textEdit_OFS.toPlainText()
512
513     if len(of) < 11:
514         self.mostrar_mensaje_error("ERROR DATOS ENTRADA", "La OF debe
515             tener 11 d gitos")
516         self.fondo_rojo(self.textEdit_OFS)
517         return False
518     else:
519         # separar n mero de OF en OV y Num Aparato.
520         # para DDPP se pueden meter hasta 6 OFS. La parte de la OV debe
521             ser igual y variar el n mero de aparato
522         # Metar las OFS separadas por enter \n
523         lista_ofs = of.split('\n')
524         if len(lista_ofs) > 6:
525             self.mostrar_mensaje_error("ERROR DATOS ENTRADA", "El
526                 m ximo de OFS es 6")
527         if self.ensayo == ENSAYO_PRECISION:
528             # Usar la primera OF
529             self.textEdit_OV.setText(lista_ofs[0][0:8])
530             self.textEdit_NA.setText(lista_ofs[0][8:])
531         elif self.ensayo == ENSAYO_DDPP:
532             self.textEdit_OV.setText(lista_ofs[0][0:8])
533             self.textEdit_NA.setText(lista_ofs[0][8:])
534
535         self.fondo_blanco(self.textEdit_OFS)
536         return True
537
538     def obtener_datos_ensayo(self):
539         """M todo para obtener los datos de ensayo.
540
541         Este abre el web service , recoge los datos de la OV y el n mero de
542             aparato y recoge los datos de SAIO o DIS3. En caso de no poder
543             hacerlo ense a un mensaje de error.
544
545         Parametros:
546             self — clase App
547
548         Atributos:
549             self.ov — Orden de venta: primeros 8 d gitos de la OF
550
551             self.num_aps — Lista de n mero de aparato: ltimos 3
552                 d gitos de la OF
553
554             self.xml — Clase ComunicacionWebService

```

```

549     Devuelve:
550         True — si se han obtenido y le do los datos de SAIO y DIS3
           correctamente
551
552         False — si ha habido alg n error
553
554     Errores:
555         Error Lectura de Datos — No se han podido leer los datos
           correctamente de las bases de datos
556     """
557     self.ov = self.textEdit_OV.toPlainText()
558     self.num_aps = self.textEdit_NA.toPlainText()
559     try:
560         self.xml.abrir_xml(self.ov, self.num_aps, self.ensayo)
561         self.xml.leer_xml_ensayo(self.ensayo)
562         self.xml.abrir_dis3(self.ov)
563         self.xml.leer_dis3()
564         return True
565     except:
566         self.show_dialog_error_lectura_datos()
567         return False
568
569     def precision(self):
570         """Funci n conectada al bot n PRECISION de la interfaz main.ui.
           Recoge en orden las funciones referentes al
571     ensayo de precisi n. Identificador: ENSAYO_PRECISION
572
573     1. Registra el tipo de ensayo: ENSAYO_PRECISION
574     2. Comprueba la selecci n del tipo de ensayo. En caso de correcto:
575         3. Muestra la informaci n de esa OF del ensayo de precisi n en
           la interfaz
576         4. Comprueba si esa OF ya est  ensayada
577
578     Par metros:
579         self — clase App
580
581     Atributos:
582         self.ensayo — tipo de ensayo seleccionado que puede ser
           precisi n o descargas parciales
583
584     """
585     self.ensayo = ENSAYO_PRECISION
586     if self.seleccion_tipo_ensayo():
587         self.mostrar_info_precision()
588         self.comprobacion_si_ensayado_precision()
589
590     def ddpp(self):
591         """Funci n conectada al bot n DDPP de la interfaz main.ui. Recoge
           en orden las funciones referentes al ensayo de descargas
           parciales. Identificador: ENSAYO_DDPP
592
593     1. Registra el tipo de ensayo: ENSAYO_DDPP
594     2. Comprueba la selecci n del tipo de ensayo. En caso de correcto:
595         3. Muestra la informaci n de esa OF del ensayo de descargas
           parciales en la interfaz
596         4. Comprueba si esa OF ya est  ensayada
597
598     Par metros:

```

```

599         self — clase App
600
601     Atributos:
602         self.ensayo — tipo de ensayo seleccionado que puede ser
603             precisi n o descargas parciales
604
605     """
606     self.ensayo = ENSAYO_DDPP
607     if self.seleccion_tipo_ensayo():
608         self.mostrar_info_ddpp()
609         self.comprobacion_si_ensayado_ddpp()
610
611 def nueva_tanda(self):
612     """Funci n para crear una nueva tanda en caso de una OF ya
613         ensayada.
614
615     1. Muestra ventana con mensaje de OF ya ensayado y pregunta si se
616         quiere una nueva tanda.
617     Opciones: Yes/No
618
619     Yes:
620         2. Muestra ventana para elegir raz n de nueva tanda. Opciones:
621             Vac o /Probado despu s de rehacer /Probado despu s de
622             reparacion/Repetici n de ensayo
623
624             Vac o:
625                 3. Ventana de error de opci n de tanda
626                 4. Vuelve a llamar a la funci n nueva_tanda
627             Para cualquier otra opci n:
628                 3. Comprueba cu l es la opci n elegida comparandola con
629                 las 3 opciones de nueva tanda.
630             Identificadores: OPCIONES_NUEVA_TANDA
631                 4. Asigna al identificador de la nueva tanda el que le
632                 corresponde.
633                 5. Lee la OV y el n mero de aparato de la interfaz
634                 6. Intenta crear nueva tanda, leer la respuesta de la base
635                 de datos y llamar a la funci n correspondiente
636                 a cada tipo de ensayo. En caso de que haya alg n error:
637                     1. Mensaje de error de nueva tanda
638                     2. Lo registra en el logger
639                     3. Vuelve a llamar a la funci n nueva_tanda
640                     7. Activa el bot n de inicio
641
642     No:
643         2. Desactiva el bot n de inicio
644
645     Par metros:
646         self — clase App
647
648     Atributos:
649         opcion_elegida — Respuesta a la pregunta de si se quiere nueva
650             tanda. Opciones: Yes/No
651
652         id_tanda_nueva — identificador de la raz n de la nueva tanda
653
654         self.ov — Orden de venta: primeros 8 d gitos de la OF
655
656         self.num_aps — Lista de n mero de aparatos: ltimos 3
657             d gitos de la OF

```

```

647
648     self.xml — Clase ComunicacionWebService
649
650     self.textEdit_OV — textEdit de la interfaz main.ui que hace
651         referencia a la casilla de OV.
652     Clase QTextEdit de la librería Qt5.
653
654     self.textEdit_NA — textEdit de la interfaz main.ui que hace
655         referencia a la casilla deL n mero de aparato.
656     Clase QTextEdit de la librería Qt5.
657
658     self.logger — Logger
659
660     """
661     if self.mostrar_mensaje_ensayado() == CLICAR_YES:
662         opcion_elegida = self.mostrar_ventana_nueva_tanda().textValue()
663         if opcion_elegida == OPCIONES_NUEVA_TANDA[0]:
664             self.mostrar_mensaje_error("Error opción de tanda",
665                                     "Es necesario especificar la
666                                     razón por la que se repite
667                                     el ensayo")
668
669             self.nueva_tanda()
670         else:
671             if opcion_elegida == OPCIONES_NUEVA_TANDA[1]:
672                 id_tanda_nueva = ID_PROBADO_REHACER
673             elif opcion_elegida == OPCIONES_NUEVA_TANDA[2]:
674                 id_tanda_nueva = ID_PROBADO_REPARACION
675             elif opcion_elegida == OPCIONES_NUEVA_TANDA[3]:
676                 id_tanda_nueva = ID_REPETICION
677             self.ov = self.textEdit_OV.toPlainText()
678             self.num_aps = int(self.textEdit_NA.toPlainText())
679             try:
680                 self.xml.crear_nueva_tanda(self.ov, self.num_aps, self.
681                                         ensayo, id_tanda_nueva)
682                 self.xml.leer_xml_ensayo(self.ensayo)
683                 if self.ensayo == ENSAYO_PRECISION:
684                     self.precision()
685                 elif self.ensayo == ENSAYO_DDPP:
686                     self.ddpp()
687             except:
688                 self.mostrar_mensaje_error("Error opción de tanda",
689                                     "El ensayo sólo puede
690                                     repetirse una vez por la
691                                     misma razón. Elija otra
692                                     opción")
693
694                 self.logger.exception("Error opción de tanda")
695                 self.nueva_tanda()
696
697         # Habilitar el botón de inicio
698         self.activar_boton_inicio()
699
700     else:
701         self.desactivar_boton_inicio()
702
703     # Funciones comenzar ensayo
704     def comenzar_conexion_hipotronics(self):
705         """Función para conectar con el Hipotronics y mandarle los datos
706         de las rampas.

```



```

696
697     1. Se conecta con el Hipotronics mediante socket. Conexi n TCP/IP.
698     2. Comprobar si se ha conectado. En caso de conexi n correcta:
699         3. Pone el Hipotronics en modo AUTO
700         4. Le manda los datos de las rampas
701         5. Le manda el trip voltage
702         6. Le manda el trip current
703     En caso de no haber conexi n:
704         3. Muestra mensaje de error de conexi n
705
706     En caso de no conectarse muestra un mensaje de error.
707
708     Par metros:
709         self — clase App
710
711     Atributos:
712         self.h — clase Hipotronics
713
714     Errores:
715         Error Conexi n HIPOTRONICS — No se ha podido establecer
716         conexi n TCP/IP con el dispositivo.
717     """
718     self.h.conectar_hipotronics()
719     if self.h.connected == 1:
720         # Configuraci n
721         self.h.set_test_mode_auto()
722         self.h.set_steps()
723         self.h.set_trip_voltage()
724         self.h.set_trip_current()
725     else:
726         self.mostrar_mensaje_error("Error conexi n hipotronics", "No
727         se ha podido conectar con el hipotronics. ")
728
729     def grabar_resultados_precision(self, punto_lectura):
730         """Funci n que graba los resultados del ensayo de precisi n ,
731         siendo los resultados las lecturas del ZERA en un punto de
732         tensi n concreto.
733
734         1. Comprueba que el n mero de lectura sea menor que la cantidad de
735         lecturas totales que tiene que realizar.
736         Si es as :
737             2. Comprueba la tensi n de esa lectura
738         Si no:
739             2. Cambia el estado a fin de ensayo. Nuevo estado :
740                 P3_FIN_ENSAYO
741
742     Par metros:
743         self — clase App
744
745         lectura — el punto de tensi n establecido en el que se hace
746         la lectura. En este momento hay 5 puntos.
747
748     Atributos:
749         self.controles_lecturas_precision — lista que hace referencia
750         a los datos de las medidas de precision
751
752         self.controles_lecturas_zera — lista que hace referencia a los
753         datos de las lecturas del puente Zera

```

```

745
746     """
747     if punto_lectura < len(self.controles_lecturas_precision):
748         self.comprobar_tension(self.controles_lecturas_zera[
749             punto_lectura + 1].toPlainText(), punto_lectura,
750             self.controles_lecturas_zera[0].
751                 toPlainText())
752
753     else:
754         self.cambiar_estado_fin_ensayo()
755
756 def configurar_zera(self):
757     """Funci n que env a la configuraci n de las tensiones y la
758     frecuencia al puente Zera.
759
760     Par metros:
761     self — clase App
762
763     Atributos:
764     self.pz — clase PuenteZera
765
766     """
767     self.pz.configurar_tensiones_evt()
768     self.pz.configurar_frecuencia()
769
770 # Funciones guardar datos
771 def guardar_datos_descargas_parciales(self):
772     """Funci n que guarda los datos del ensayo de descargas parciales
773     en la base de datos, SAIO.
774
775     1. Recoge los valores del ensayo de descargas parciales escritos en
776     la interfaz.
777
778     Intenta:
779     2. Guardar los datos recogidos en SAIO.
780     3. Comprobar si hay m s de un aparato. En caso de ser as se
781     abre SAIO para cada n mero de aparato
782     y se guardan los valores del ensayo en todas ellas. Estos
783     valores son comunes para todos.
784     4. Inicializar los datos de entrada
785     5. Limpia los datos del ensayo de la interfaz
786     6. Muestra mensaje de que el ensayo se ha guardado
787     correctamente
788
789     Si hay alg n error:
790     2. Muestra mensaje de error de guardado
791     3. Se guarda el error en el logger
792
793     Par metros:
794     self — clase App
795
796     Atributos:
797     valor_ddp_1 — valor escrito en la interfaz del valor del
798     ensayo de descargas parciales en el primer punto
799
800     valor_ddp_2 — valor escrito en la interfaz del valor del
801     ensayo de descargas parciales en el segundo punto

```

```

792     valor_ddp_3 — valor escrito en la interfaz del valor del
793         ensayo de descargas parciales en el tercer punto
794     valor_ddp_4 — valor escrito en la interfaz del valor del
795         ensayo de descargas parciales en el cuarto punto
796     valor_ddp_5 — valor escrito en la interfaz del valor del
797         ensayo de descargas parciales en el primer punto
798     self.textEdit_ddpp1 — textEdit de la interfaz main.ui que hace
799     de descargas parciales en el primer punto. Clase QTextEdit de
800     la librería Qt5.
801     self.textEdit_ddpp2 — textEdit de la interfaz main.ui que hace
802     de descargas parciales en el segundo punto. Clase QTextEdit de
803     la librería Qt5.
804     self.textEdit_ddpp3 — textEdit de la interfaz main.ui que hace
805     de descargas parciales en el tercer punto. Clase QTextEdit de
806     la librería Qt5.
807     self.textEdit_ddpp4 — textEdit de la interfaz main.ui que hace
808     de descargas parciales en el cuarto punto. Clase QTextEdit de
809     la librería Qt5.
810     self.textEdit_ddpp5 — textEdit de la interfaz main.ui que hace
811     de descargas parciales en el quinto punto. Clase QTextEdit de
812     la librería Qt5.
813     self.xml — clase ComunicacionWebService
814     self.num_aps — lista de número de aparatos: últimos 3
815         dígitos de la OF
816     self.ov — Orden de venta: primeros 8 dígitos de la OF
817     self.ensayo — tipo de ensayo seleccionado que puede ser
818         preciso o descargas parciales
819     logger — Logger
820
821     Errores:
822     Error Guardar — No se han podido guardar los datos en la base
823     de datos.
824
825     """
826     valor_ddp_1 = self.textEdit_ddpp1.toPlainText()
827     valor_ddp_2 = self.textEdit_ddpp2.toPlainText()
828     valor_ddp_3 = self.textEdit_ddpp3.toPlainText()
829     valor_ddp_4 = self.textEdit_ddpp4.toPlainText()
830     valor_ddp_5 = self.textEdit_ddpp5.toPlainText()
831
832     try:

```

```

834         self.xml.guardar_xml_ddpp(valor_ddp_1, valor_ddp_2, valor_ddp_3
835             , valor_ddp_4, valor_ddp_5)
836
837         # si se ha indicado m s de un aparato. guardar el ensayo en
838         # los otros aparatos
839         if len(self.num_aps) > 1:
840             for i, num in enumerate(self.num_aps):
841                 if i > 0:
842                     self.xml.abrir_xml(self.ov, num, self.ensayo)
843                     self.xml.guardar_xml_ddpp(valor_ddp_1, valor_ddp_2,
844                         valor_ddp_3, valor_ddp_4, valor_ddp_5)
845
846         # cambiar interfaz
847         self.inicializar_datos_entrada()
848         self.limpiar_datos_ddpp()
849         self.mostrar_mensaje_error("GUARDAR ENSAYO", "El ensayo se ha
850             guardado correctamente")
851     except:
852         self.mostrar_mensaje_error("GUARDAR ENSAYO", "Error al guardar
853             ensayo")
854         self.logger.exception("Error al guardar ensayo")
855
856 def guardar_datos_precision(self):
857     """Funci n que guarda los datos del ensayo de precision en la base
858     de datos, SAIO.
859
860     1. Recoge los valores del ensayo de precision escritos en la
861     interfaz. Tanto los valores del error en m dulo como los
862     valores del desfase.
863
864     Intenta:
865     2. Guardar los datos recogidos en SAIO.
866     3. Comprobar si hay m s de un aparato. En caso de ser as se
867     abre SAIO para cada n mero de aparato y se guardan los
868     valores del ensayo en todas ellas. Estos valores son comunes
869     para todos.
870     4. Inicializar los datos de entrada
871     5. Limpia los datos del ensayo de la interfaz
872     6. Muestra mensaje de que el ensayo se ha guardado
873     correctamente
874
875     Si hay alg n error:
876     2. Muestra mensaje de error de guardado
877     3. Se guarda el error en el logger
878
879     Par metros:
880     self — clase App
881
882     Atributos:
883     error_modulo1 — valor escrito en la interfaz del error en
884     m dulo del ensayo de precision en el primer punto.
885
886     error_modulo2 — valor escrito en la interfaz del error en
887     m dulo del ensayo de precision en el segundo punto.
888
889     error_modulo3 — valor escrito en la interfaz del error en
890     m dulo del ensayo de precision en el tercer punto.

```

876 *error\_modulo4* — valor escrito en la interfaz del error en  
m dulo del ensayo de precision en el cuarto punto.

877

878 *error\_modulo5* — valor escrito en la interfaz del error en  
m dulo del ensayo de precision en el quinto punto.

879

880 *error\_angulo1* — valor escrito en la interfaz del error en  
ngulo del ensayo de precision en el primer punto.

881

882 *error\_angulo2* — valor escrito en la interfaz del error en  
ngulo del ensayo de precision en el segundo punto.

883

884 *error\_angulo3* — valor escrito en la interfaz del error en  
ngulo del ensayo de precision en el tercer punto.

885

886 *error\_angulo4* — valor escrito en la interfaz del error en  
ngulo del ensayo de precision en el cuarto punto.

887

888 *error\_angulo5* — valor escrito en la interfaz del error en  
ngulo del ensayo de precision en el quinto punto.

889

890 *self.label\_Error\_Modulo\_2* — label de la interfaz *main.ui* que  
hace referencia a la casilla del valor del  
891 error en m dulo del ensayo de precision en el primer punto.  
Clase *QLabel* de la librer a *Qt5*.

892

893 *self.label\_Error\_Modulo\_3* — label de la interfaz *main.ui* que  
hace referencia a la casilla del valor del  
894 error en m dulo del ensayo de precision en el segundo punto.  
Clase *QLabel* de la librer a *Qt5*.

895

896 *self.label\_Error\_Modulo\_4* — label de la interfaz *main.ui* que  
hace referencia a la casilla del valor del  
897 error en m dulo del ensayo de precision en el tercer punto.  
Clase *QLabel* de la librer a *Qt5*.

898

899 *self.label\_Error\_Modulo\_5* — label de la interfaz *main.ui* que  
hace referencia a la casilla del valor del error en m dulo  
del ensayo de precision en el cuarto punto. Clase *QLabel* de  
la librer a *Qt5*.

900

901 *self.label\_Error\_Modulo\_6* — label de la interfaz *main.ui* que  
hace referencia a la casilla del valor del error en m dulo  
del ensayo de precision en el quinto punto. Clase *QLabel* de  
la librer a *Qt5*.

902

903 *self.label\_desfase\_2* — label de la interfaz *main.ui* que hace  
referencia a la casilla del valor del desfase del ensayo de  
precision en el primer punto. Clase *QLabel* de la librer a  
*Qt5*.

904

905 *self.label\_desfase\_3* — label de la interfaz *main.ui* que hace  
referencia a la casilla del valor del desfase del ensayo de  
precision en el segundo punto. Clase *QLabel* de la librer a  
*Qt5*.

906

907 *self.label\_desfase\_4* — label de la interfaz *main.ui* que hace  
referencia a la casilla del valor del desfase del ensayo de

```

    precision en el tercer punto. Clase QLabel de la libreria a
    Qt5.
908
909 self.label_desfase_5 — label de la interfaz main.ui que hace
    referencia a la casilla del valor del desfase del ensayo de
    precision en el cuarto punto. Clase QLabel de la libreria a
    Qt5.
910
911 self.label_desfase_6 — label de la interfaz main.ui que hace
    referencia a la casilla del valor del desfase del ensayo de
    precision en el quinto punto. Clase QLabel de la libreria a
    Qt5.
912
913 self.xml — clase ComunicacionWebService
914
915 self.num_aps — lista de numero de aparatos: ltimos 3
    d gitos de la OF
916
917 self.ov — Orden de venta: primeros 8 d gitos de la OF
918
919 self.ensayo — tipo de ensayo seleccionado que puede ser
    precisi n o descargas parciales
920
921 logger — Logger
922
923 Errores:
924 Error Guardar — No se han podido guardar los datos en la base
    de datos.
925
926 """
927 error_modulo1 = self.label_Error_Modulo_2.text()
928 error_modulo2 = self.label_Error_Modulo_3.text()
929 error_modulo3 = self.label_Error_Modulo_4.text()
930 error_modulo4 = self.label_Error_Modulo_5.text()
931 error_modulo5 = self.label_Error_Modulo_6.text()
932 #
933 error_angulo1 = self.label_desfase_2.text()
934 error_angulo2 = self.label_desfase_3.text()
935 error_angulo3 = self.label_desfase_4.text()
936 error_angulo4 = self.label_desfase_5.text()
937 error_angulo5 = self.label_desfase_6.text()
938
939 try:
940     self.xml.guardar_xml(error_modulo1, error_modulo2,
        error_modulo3, error_modulo4,
941         error_modulo5, error_angulo1,
        error_angulo2, error_angulo3,
942         error_angulo4, error_angulo5)
943     # si se ha indicado m s de un aparato. guardar el ensayo en
        los otros aparatos
944     if len(self.num_aps) > 1:
945         for i, num in enumerate(self.num_aps):
946             if i > 0:
947                 self.xml.abrir_xml(self.ov, num, self.ensayo)
948                 self.xml.guardar_xml(error_modulo1, error_modulo2,
        error_modulo3, error_modulo4,
949                     error_modulo5, error_angulo1,
        error_angulo2,

```

```

950         error_angulo3 ,
951         error_angulo4 , error_angulo5)
952     # interfaz
953     self.inicializar_datos_entrada()
954     self.limpiar_datos_precision()
955     self.mostrar_mensaje_error("GUARDAR ENSAYO", "El ensayo se ha
956         guardado correctamente")
957
958 except:
959     self.mostrar_mensaje_error("GUARDAR ENSAYO", "Error al guardar
960         ensayo")
961     self.logger.exception("Error al guardar ensayo")
962
963 # Funciones de cambios de estado
964 def cambiar_estado_log_in(self):
965     """Funci n que cambia el estado en el que se encuentra la m quina
966         de estados.
967
968     Nuevo estado —> P0_LOG_IN
969
970     Par metros:
971     self — clase App
972
973     Atributos:
974     self.maquina_estados — clase MaquinaEstados
975
976     """
977     self.maquina_estados.cambiar_estado(self.maquina_estados.Estados.
978         P0_LOG_IN)
979
980 def cambiar_estado_seleccion_tipo_ensayo(self):
981     """Funci n que cambia el estado en el que se encuentra la m quina
982         de estados.
983
984     Nuevo estado —> P1_SELECCION_TIPO_ENSAYOS
985
986     Par metros:
987     self — clase App
988
989     Atributos:
990     self.maquina_estados — clase MaquinaEstados
991
992     """
993     self.maquina_estados.cambiar_estado(self.maquina_estados.Estados.
994         P1_SELECCION_TIPO_ENSAYOS)
995
996 def cambiar_estado_comenzar_ensayo(self):
997     """Funci n que cambia el estado en el que se encuentra la m quina
998         de estados.
999
1000     Nuevo estado —> P2_COMENZAR_ENSAYO
1001
1002     Par metros:
1003     self — clase App
1004
1005     Atributos:
1006     self.maquina_estados — clase MaquinaEstados

```

```

1000     """
1001     self.maquina_estados.cambiar_estado(self.maquina_estados.Estados.
        P2_COMENZAR_ENSAYO)
1002
1003     def cambiar_estado_fin_ensayo(self):
1004         """Funci n que cambia el estado en el que se encuentra la m quina
        de estados.
1005
1006         Nuevo estado → P3_FIN_ENSAYO
1007
1008         Par metros:
1009             self — clase App
1010
1011         Atributos:
1012             self.maquina_estados — clase MaquinaEstados
1013
1014         """
1015     self.maquina_estados.cambiar_estado(self.maquina_estados.Estados.
        P3_FIN_ENSAYO)
1016
1017     def cambiar_estado_parar(self):
1018         """Funci n que cambia el estado en el que se encuentra la m quina
        de estados.
1019
1020         Nuevo estado → P4_PARAR_ENSAYO
1021
1022         Par metros:
1023             self — clase App
1024
1025         Atributos:
1026             self.maquina_estados — clase MaquinaEstados
1027
1028         """
1029     self.maquina_estados.cambiar_estado(self.maquina_estados.Estados.
        P4_PARAR_ENSAYO)
1030
1031     def cambiar_estado_grabar(self):
1032         """Funci n que cambia el estado en el que se encuentra la m quina
        de estados.
1033
1034         Nuevo estado → P5_GRABAR_DATOS
1035
1036         Par metros:
1037             self — clase App
1038
1039         Atributos:
1040             self.maquina_estados — clase MaquinaEstados
1041
1042         """
1043     self.maquina_estados.cambiar_estado(self.maquina_estados.Estados.
        P5_GRABAR_DATOS)
1044
1045     def cambiar_estado_descartar(self):
1046         """Funci n que cambia el estado en el que se encuentra la m quina
        de estados.
1047
1048         Nuevo estado → P6_DESCARTAR_DATOS
1049

```



```

1050     Par metros:
1051         self — clase App
1052
1053     Atributos:
1054         self.maquina_estados — clase MaquinaEstados
1055
1056     """
1057     self.maquina_estados.cambiar_estado(self.maquina_estados.Estados.
1058         P6_DESCARTAR_DATOS)
1059
1060 def cambiar_estado_configurar_zera(self):
1061     """Funci n que cambia el estado en el que se encuentra la m quina
1062     de estados.
1063
1064     Nuevo estado —> P8_CONFIGURAR_ZERA
1065
1066     Par metros:
1067         self — clase App
1068
1069     Atributos:
1070         self.maquina_estados — clase MaquinaEstados
1071
1072     """
1073     self.maquina_estados.cambiar_estado(self.maquina_estados.Estados.
1074         P8_CONFIGURAR_ZERA)
1075
1076 def cambiar_estado_grabar_resultados_interfaz(self):
1077     """Funci n que cambia el estado en el que se encuentra la m quina
1078     de estados.
1079
1080     Nuevo estado —> P9_GRABAR_RESULTADO_INTERFAZ
1081
1082     Par metros:
1083         self — clase App
1084
1085     Atributos:
1086         self.maquina_estados — clase MaquinaEstados
1087
1088     """
1089     self.maquina_estados.cambiar_estado(self.maquina_estados.Estados.
1090         P9_GRABAR_RESULTADO_INTERFAZ)
1091
1092 # Funciones generales
1093 def crear_grupos_controles(self):
1094     """Funci n que agrupa los QTextEdit y los QLabel de la interfaz
1095     creada con Qt5 en listas seg n su funci n.
1096
1097     Par metros:
1098         self — clase App
1099
1100     Atributos:
1101     Grupos de controles de la interfaz:
1102         self.controles_datos_entrada — lista que hace referencia a
1103         los datos de entrada
1104
1105         self.controles_hipotronics — lista que hace referencia a
1106         los datos del Hipotronics

```

1100 *self.controles\_lecturas\_zera* — lista que hace referencia a los datos de las lecturas del puente Zera

1101

1102 *self.controles\_configuracion\_zera* — lista que hace referencia a los datos de la configuración del puente Zera

1103

1104 *self.controles\_info\_general* — lista que hace referencia a la información general de la OF

1105

1106 *self.controles\_lecturas\_ddpp* — lista que hace referencia a los datos de las medidas de descargas parciales

1107

1108 *self.controles\_lecturas\_precision* — lista que hace referencia a los datos de las medidas de precisión

1109

1110 *self.controles\_limite\_negativo\_angulo* — lista que hace referencia a los límites negativos del error en ángulo para ensayo de precisión

1111

1112 *self.controles\_limite\_negativo\_modulo* — lista que hace referencia a los límites negativos del error en módulo para ensayo de precisión

1113

1114 *self.controles\_limite\_positivo\_angulo* — lista que hace referencia a los límites positivos del error en ángulo para ensayo de precisión

1115

1116 *self.controles\_limite\_positivo\_modulo* — lista que hace referencia a los límites positivos del error en módulo para ensayo de precisión

1117

1118 *self.controles\_limites\_ddpp* — lista que hace referencia a los límites del error en descargas parciales

1119

1120 *self.controles\_valores\_error\_modulo* — lista que hace referencia a los errores en módulo para ensayo de precisión

1121

1122 *self.controles\_valores\_error\_angulo* — lista que hace referencia a los errores en ángulo para ensayo de precisión

1123

1124

1125 *QTextEdit:*

1126 *self.textEdit\_OFS* — *QTextEdit* de la interfaz *main.ui* que hace referencia a la casilla de OF. Clase *QTextEdit* de la librería Qt5.

1127

1128 *self.textEdit\_OV* — *QTextEdit* de la interfaz *main.ui* que hace referencia a la casilla de OV. Clase *QTextEdit* de la librería Qt5.

1129

1130 *self.textEdit\_NA* — *QTextEdit* de la interfaz *main.ui* que hace referencia a la casilla del número de aparato. Clase *QTextEdit* de la librería Qt5.

1131

1132 *self.textEdit\_Np* — *QTextEdit* de la interfaz *main.ui* que hace referencia a la casilla de la tensión primaria del

1133                    *patr n. Clase QTextEdit de la librer a Qt5.*

1134                    *self.textEdit\_Ns — textEdit de la interfaz main.ui que*  
                      *hace referencia a la casilla de la tensi n secundaria*  
                      *del patr n. Clase QTextEdit de la librer a Qt5.*

1135                    *self.textEdit\_EVTp — textEdit de la interfaz main.ui que*  
1136                    *hace referencia a la casilla de la tensi n primaria de*  
                      *la tensi n a medir. Clase QTextEdit de la librer a Qt5.*

1137                    *self.textEdit\_EVTs — textEdit de la interfaz main.ui que*  
1138                    *hace referencia a la casilla de la tensi n secundaria*  
                      *de la tensi n a medir. Clase QTextEdit de la librer a*  
                      *Qt5.*

1139                    *self.textEdit\_freq — textEdit de la interfaz main.ui que*  
1140                    *hace referencia a la casilla de la frecuencia. Clase*  
                      *QTextEdit de la librer a Qt5.*

1141                    *self.textEdit\_horquilla — textEdit de la interfaz main.ui*  
1142                    *que hace referencia a la casilla de la*  
1143                    *horquilla. Clase QTextEdit de la librer a Qt5.*

1144                    *self.textEdit\_L1 — textEdit de la interfaz main.ui que*  
1145                    *hace referencia a la casilla del primer punto de*  
                      *tensi n a medir en %. Clase QTextEdit de la librer a*  
                      *Qt5.*

1146                    *self.textEdit\_L2 — textEdit de la interfaz main.ui que*  
1147                    *hace referencia a la casilla del segundo punto de*  
                      *tensi n a medir en %. Clase QTextEdit de la librer a*  
                      *Qt5.*

1148                    *self.textEdit\_L3 — textEdit de la interfaz main.ui que*  
1149                    *hace referencia a la casilla del tercer punto de*  
                      *tensi n a medir en %. Clase QTextEdit de la librer a*  
                      *Qt5.*

1150                    *self.textEdit\_L4 — textEdit de la interfaz main.ui que*  
1151                    *hace referencia a la casilla del cuarto punto de*  
                      *tensi n a medir en %. Clase QTextEdit de la librer a*  
                      *Qt5.*

1152                    *self.textEdit\_L5 — textEdit de la interfaz main.ui que*  
1153                    *hace referencia a la casilla del quinto punto de*  
                      *tensi n a medir en %. Clase QTextEdit de la librer a*  
                      *Qt5.*

1154                    *self.textEdit\_Rampa1 — textEdit de la interfaz main.ui que*  
1155                    *hace referencia a la casilla del valor de la*  
                      *inclinaci n de la rampa del Hipotronics del primer*  
                      *punto. Clase QTextEdit de la librer a Qt5.*

1156                    *self.textEdit\_Rampa2 — textEdit de la interfaz main.ui que*  
1157                    *hace referencia a la casilla del valor de la*  
                      *inclinaci n de la rampa del Hipotronics del segundo*  
                      *punto. Clase QTextEdit de la librer a Qt5.*

1158

1159 *self.textEdit\_Rampa3* — *textEdit* de la interfaz *main.ui* que  
1160 hace referencia a la casilla del valor de la  
inclinaci n de la rampa del Hipotronics del tercer punto.  
Clase *QTextEdit* de la librer a Qt5.

1161 *self.textEdit\_Rampa4* — *textEdit* de la interfaz *main.ui* que  
1162 hace referencia a la casilla del valor de la  
inclinaci n de la rampa del Hipotronics del cuarto  
punto. Clase *QTextEdit* de la librer a Qt5.

1163 *self.textEdit\_Rampa5* — *textEdit* de la interfaz *main.ui* que  
1164 hace referencia a la casilla del valor de la  
inclinaci n de la rampa del Hipotronics del quinto  
punto. Clase *QTextEdit* de la librer a Qt5.

1165 *self.textEdit\_Tension1* — *textEdit* de la interfaz *main.ui*  
1166 que hace referencia a la casilla del valor de la  
tensi n de la rampa del Hipotronics en el primer punto.  
Clase *QTextEdit* de la librer a Qt5.

1167 *self.textEdit\_Tension2* — *textEdit* de la interfaz *main.ui*  
1168 que hace referencia a la casilla del valor de la  
tensi n de la rampa del Hipotronics en el segundo punto  
. Clase *QTextEdit* de la librer a Qt5.

1169 *self.textEdit\_Tension3* — *textEdit* de la interfaz *main.ui*  
1170 que hace referencia a la casilla del valor de la  
tensi n de la rampa del Hipotronics en el tercer punto.  
Clase *QTextEdit* de la librer a Qt5.

1171 *self.textEdit\_Tension4* — *textEdit* de la interfaz *main.ui*  
1172 que hace referencia a la casilla del valor de la  
tensi n de la rampa del Hipotronics en el cuarto punto.  
Clase *QTextEdit* de la librer a Qt5.

1173 *self.textEdit\_Tension5* — *textEdit* de la interfaz *main.ui*  
1174 que hace referencia a la casilla del valor de la  
tensi n de la rampa del Hipotronics en el quinto punto.  
Clase *QTextEdit* de la librer a Qt5.

1175 *self.textEdit\_Tiempo1* — *textEdit* de la interfaz *main.ui*  
1176 que hace referencia a la casilla del tiempo de espera de  
la rampa del Hipotronics en el primer punto. Clase  
*QTextEdit* de la librer a Qt5.

1177 *self.textEdit\_Tiempo2* — *textEdit* de la interfaz *main.ui*  
1178 que hace referencia a la casilla del tiempo de espera de  
la rampa del Hipotronics en el segundo punto. Clase  
*QTextEdit* de la librer a Qt5.

1179 *self.textEdit\_Tiempo3* — *textEdit* de la interfaz *main.ui*  
1180 que hace referencia a la casilla del tiempo de espera de  
la rampa del Hipotronics en el tercer punto. Clase  
*QTextEdit* de la librer a Qt5.

1181 *self.textEdit\_Tiempo4* — *textEdit* de la interfaz *main.ui*  
1182 que hace referencia a la casilla del tiempo de espera de  
la rampa del Hipotronics en el cuarto punto. Clase

1183 *QTextEdit de la librer a Qt5.*

1184 *self.textEdit\_Tiempo5 — textEdit de la interfaz main.ui*  
*que hace referencia a la casilla del tiempo de espera de*  
*la rampa del Hipotronics en el quinto punto. Clase*  
*QTextEdit de la librer a Qt5.*

1185

1186 *self.textEdit\_ddpp1 — textEdit de la interfaz main.ui que*  
*hace referencia a la casilla del valor del ensayo*  
1187 *de descargas parciales en el primer punto. Clase QTextEdit*  
*de la librer a Qt5.*

1188

1189 *self.textEdit\_ddpp2 — textEdit de la interfaz main.ui que*  
*hace referencia a la casilla del valor del ensayo*  
1190 *de descargas parciales en el segundo punto. Clase QTextEdit*  
*de la librer a Qt5.*

1191

1192 *self.textEdit\_ddpp3 — textEdit de la interfaz main.ui que*  
*hace referencia a la casilla del valor del ensayo*  
1193 *de descargas parciales en el tercer punto. Clase QTextEdit*  
*de la librer a Qt5.*

1194

1195 *self.textEdit\_ddpp4 — textEdit de la interfaz main.ui que*  
*hace referencia a la casilla del valor del ensayo*  
1196 *de descargas parciales en el cuarto punto. Clase QTextEdit*  
*de la librer a Qt5.*

1197

1198 *self.textEdit\_ddpp5 — textEdit de la interfaz main.ui que*  
*hace referencia a la casilla del valor del ensayo*  
1199 *de descargas parciales en el quinto punto. Clase QTextEdit*  
*de la librer a Qt5.*

1200

1201 *QLabel:*

1202 *self.label\_columna\_titulos1 — label de la interfaz main.ui*  
*que hace referencia a la primera columna de t tulos de*  
*la informaci n general. Clase QLabel de la librer a*  
*Qt5.*

1203

1204 *self.label\_columna\_titulos2 — label de la interfaz main.ui*  
*que hace referencia a la segunda columna de t tulos de*  
*la informaci n general. Clase QLabel de la librer a*  
*Qt5.*

1205

1206 *self.label\_columna\_valores1 — label de la interfaz main.ui*  
*que hace referencia a la primera columna de valores de*  
*la informaci n general. Clase QLabel de la librer a*  
*Qt5.*

1207

1208 *self.label\_columna\_valores2 — label de la interfaz main.ui*  
*que hace referencia a la segunda columna de valores de*  
*la informaci n general. Clase QLabel de la librer a*  
*Qt5.*

1209

1210 *self.label\_limite\_ddpp1 — label de la interfaz main.ui que*  
*hace referencia al limite del valor del ensayo de*  
*descargas parciales en el primer punto. Clase QLabel de*  
*la librer a Qt5.*

1211

1212 *self.label\_limite\_ddpp2* — label de la interfaz *main.ui* que  
hace referencia al limite del valor del ensayo de  
descargas parciales en el segundo punto. Clase *QLabel* de  
la libreria *Qt5*.

1213  
1214 *self.label\_limite\_ddpp3* — label de la interfaz *main.ui* que  
hace referencia al limite del valor del ensayo de  
descargas parciales en el tercer punto. Clase *QLabel* de  
la libreria *Qt5*.

1215  
1216 *self.label\_limite\_ddpp4* — label de la interfaz *main.ui* que  
hace referencia al limite del valor del ensayo de  
descargas parciales en el cuarto punto. Clase *QLabel* de  
la libreria *Qt5*.

1217  
1218 *self.label\_limite\_ddpp5* — label de la interfaz *main.ui* que  
hace referencia al limite del valor del ensayo  
1219 de descargas parciales en el quinto punto. Clase *QLabel* de  
la libreria *Qt5*.

1220  
1221 *self.label\_LPN\_2* — label de la interfaz *main.ui* que hace  
referencia al valor de la tensi n medida en el primer  
punto del ensayo de precisi n en %. Clase *QLabel* de la  
libreria *Qt5*.

1222  
1223 *self.label\_LPN\_3* — label de la interfaz *main.ui* que hace  
referencia al valor de la tensi n medida en el segundo  
punto del ensayo de precisi n en %. Clase *QLabel* de la  
libreria *Qt5*.

1224  
1225 *self.label\_LPN\_4* — label de la interfaz *main.ui* que hace  
referencia al valor de la tensi n medida en el tercer  
punto del ensayo de precisi n en %. Clase *QLabel* de la  
libreria *Qt5*.

1226  
1227 *self.label\_LPN\_5* — label de la interfaz *main.ui* que hace  
referencia al valor de la tensi n medida en el cuarto  
punto del ensayo de precisi n en %. Clase *QLabel* de la  
libreria *Qt5*.

1228  
1229 *self.label\_LPN\_6* — label de la interfaz *main.ui* que hace  
referencia al valor de la tensi n medida en el quinto  
punto del ensayo de precisi n en %. Clase *QLabel* de la  
libreria *Qt5*.

1230  
1231 *self.label\_Error\_Modulo\_2* — label de la interfaz *main.ui*  
que hace referencia al valor del error en m dulo medido  
en el primer punto del ensayo de precisi n en %. Clase  
*QLabel* de la libreria *Qt5*.

1232  
1233 *self.label\_Error\_Modulo\_3* — label de la interfaz *main.ui*  
que hace referencia al valor del error en m dulo medido  
en el segundo punto del ensayo de precisi n en %.  
Clase *QLabel* de la libreria *Qt5*.

1234  
1235 *self.label\_Error\_Modulo\_4* — label de la interfaz *main.ui*  
que hace referencia al valor del error en m dulo medido  
en el tercer punto del ensayo de precisi n en %. Clase

1236 *QLabel de la librería Qt5.*

1237 *self.label\_Error\_Modulo\_5 — label de la interfaz main.ui que hace referencia al valor del error en módulo medido en el cuarto punto del ensayo de precisión en %. Clase QLabel de la librería Qt5.*

1238

1239 *self.label\_Error\_Modulo\_6 — label de la interfaz main.ui que hace referencia al valor del error en módulo medido en el quinto punto del ensayo de precisión en %. Clase QLabel de la librería Qt5.*

1240

1241 *self.label\_limite\_negativo\_modulo1 — label de la interfaz main.ui que hace referencia al límite negativo*

1242 *del error en módulo del ensayo de descargas parciales en el primer punto. Clase QLabel de la librería Qt5.*

1243

1244 *self.label\_limite\_negativo\_modulo2 — label de la interfaz main.ui que hace referencia al límite negativo*

1245 *del error en módulo del ensayo de descargas parciales en el segundo punto. Clase QLabel de la librería Qt5.*

1246

1247 *self.label\_limite\_negativo\_modulo3 — label de la interfaz main.ui que hace referencia al límite negativo*

1248 *del error en módulo del ensayo de descargas parciales en el tercer punto. Clase QLabel de la librería Qt5.*

1249

1250 *self.label\_limite\_negativo\_modulo4 — label de la interfaz main.ui que hace referencia al límite negativo*

1251 *del error en módulo del ensayo de descargas parciales en el cuarto punto. Clase QLabel de la librería Qt5.*

1252

1253 *self.label\_limite\_negativo\_modulo5 — label de la interfaz main.ui que hace referencia al límite negativo*

1254 *del error en módulo del ensayo de descargas parciales en el quinto punto. Clase QLabel de la librería Qt5.*

1255

1256 *self.label\_limite\_positivo\_modulo1 — label de la interfaz main.ui que hace referencia al límite positivo*

1257 *del error en módulo del ensayo de descargas parciales en el primer punto. Clase QLabel de la librería Qt5.*

1258

1259 *self.label\_limite\_positivo\_modulo2 — label de la interfaz main.ui que hace referencia al límite positivo*

1260 *del error en módulo del ensayo de descargas parciales en el segundo punto. Clase QLabel de la librería Qt5.*

1261

1262 *self.label\_limite\_positivo\_modulo3 — label de la interfaz main.ui que hace referencia al límite positivo*

1263 *del error en módulo del ensayo de descargas parciales en el tercer punto. Clase QLabel de la librería Qt5.*

1264

1265 *self.label\_limite\_positivo\_modulo4 — label de la interfaz main.ui que hace referencia al límite positivo*

1266 *del error en módulo del ensayo de descargas parciales en el cuarto punto. Clase QLabel de la librería Qt5.*

1267



1268 *self.label\_limite\_positivo\_modulo5* — label de la interfaz  
1269 *main.ui* que hace referencia al limite positivo  
del error en modulo del ensayo de descargas parciales en  
el quinto punto. Clase *QLabel* de la libreria *Qt5*.

1270  
1271 *self.label\_desfase\_2* — label de la interfaz *main.ui* que  
hace referencia al valor del desfase medido en  
1272 el primer punto del ensayo de precision en %. Clase *QLabel*  
de la libreria *Qt5*.

1273  
1274 *self.label\_desfase\_3* — label de la interfaz *main.ui* que  
hace referencia al valor del desfase medido en  
1275 el segundo punto del ensayo de precision en %. Clase  
*QLabel* de la libreria *Qt5*.

1276  
1277 *self.label\_desfase\_4* — label de la interfaz *main.ui* que  
hace referencia al valor del desfase medido en  
1278 el tercer punto del ensayo de precision en %. Clase *QLabel*  
de la libreria *Qt5*.

1279  
1280 *self.label\_desfase\_5* — label de la interfaz *main.ui* que  
hace referencia al valor del desfase medido en  
1281 el cuarto punto del ensayo de precision en %. Clase *QLabel*  
de la libreria *Qt5*.

1282  
1283 *self.label\_desfase\_6* — label de la interfaz *main.ui* que  
hace referencia al valor del desfase medido en  
1284 el quinto punto del ensayo de precision en %. Clase *QLabel*  
de la libreria *Qt5*.

1285  
1286 *self.label\_limite\_negativo\_desfase1* — label de la interfaz  
*main.ui* que hace referencia al limite negativo  
1287 del desfase del ensayo de descargas parciales en el primer  
punto. Clase *QLabel* de la libreria *Qt5*.

1288  
1289 *self.label\_limite\_negativo\_desfase2* — label de la interfaz  
*main.ui* que hace referencia al limite negativo  
1290 del desfase del ensayo de descargas parciales en el segundo  
punto. Clase *QLabel* de la libreria *Qt5*.

1291  
1292 *self.label\_limite\_negativo\_desfase3* — label de la interfaz  
*main.ui* que hace referencia al limite negativo  
1293 del desfase del ensayo de descargas parciales en el tercer  
punto. Clase *QLabel* de la libreria *Qt5*.

1294  
1295 *self.label\_limite\_negativo\_desfase4* — label de la interfaz  
*main.ui* que hace referencia al limite negativo  
1296 del desfase del ensayo de descargas parciales en el cuarto  
punto. Clase *QLabel* de la libreria *Qt5*.

1297  
1298 *self.label\_limite\_negativo\_desfase5* — label de la interfaz  
*main.ui* que hace referencia al limite negativo  
1299 del desfase del ensayo de descargas parciales en el quinto  
punto. Clase *QLabel* de la libreria *Qt5*.

1300  
1301 *self.label\_limite\_positivodesfase\_desfase1* — label de la  
interfaz *main.ui* que hace referencia al limite positivo



```

1302         del desfase del ensayo de descargas parciales en el primer
           punto. Clase QLabel de la librer a Qt5.
1303
1304         self.label_limite_positivo_desfase2 — label de la interfaz
           main.ui que hace referencia al limite positivo
1305         del desfase del ensayo de descargas parciales en el segundo
           punto. Clase QLabel de la librer a Qt5.
1306
1307         self.label_limite_positivo_desfase3 — label de la interfaz
           main.ui que hace referencia al limite positivo
1308         del desfase del ensayo de descargas parciales en el tercer
           punto. Clase QLabel de la librer a Qt5.
1309
1310         self.label_limite_positivo_desfase4 — label de la interfaz
           main.ui que hace referencia al limite positivo
1311         del desfase del ensayo de descargas parciales en el cuarto
           punto. Clase QLabel de la librer a Qt5.
1312
1313         self.label_limite_positivo_desfase5 — label de la interfaz
           main.ui que hace referencia al limite positivo
1314         del desfase del ensayo de descargas parciales en el quinto
           punto. Clase QLabel de la librer a Qt5.
1315     """
1316     self.controles_datos_entrada = [self.textEdit_OFS, self.textEdit_OV
           , self.textEdit_NA]
1317     self.controles_configuracion_zera = [self.textEdit_Np, self.
           textEdit_Ns, self.textEdit_EVTp, self.textEdit_EVTs, self.
           textEdit_freq]
1318     self.controles_lecturas_zera = [self.textEdit_horquilla, self.
           textEdit_L1, self.textEdit_L2, self.textEdit_L3, self.
           textEdit_L4, self.textEdit_L5]
1319     self.controles_info_general = [self.label_columna_titulos1, self.
           label_columna_titulos2, self.label_columna_valores1, self.
           label_columna_valores2]
1320     self.controles_hipotronics = [self.textEdit_Rampa1, self.
           textEdit_Rampa2, self.textEdit_Rampa3, self.textEdit_Rampa4, self.
           textEdit_Rampa5, self.textEdit_Tension1, self.textEdit_Tension2
           , self.textEdit_Tension3, self.textEdit_Tension4, self.
           textEdit_Tension5, self.textEdit_Tiempo1, self.textEdit_Tiempo2,
           self.textEdit_Tiempo3, self.textEdit_Tiempo4, self.
           textEdit_Tiempo5]
1321     self.controles_lecturas_ddpp = [self.textEdit_ddpp1, self.
           textEdit_ddpp2, self.textEdit_ddpp3, self.textEdit_ddpp4, self.
           textEdit_ddpp5]
1322     self.controles_limites_ddpp = [self.label_limite_ddpp1, self.
           label_limite_ddpp2, self.label_limite_ddpp3, self.
           label_limite_ddpp4, self.label_limite_ddpp5]
1323     self.controles_lecturas_precision = [self.label_LPN_2, self.
           label_LPN_3, self.label_LPN_4, self.label_LPN_5, self.
           label_LPN_6]
1324     self.controles_valores_error_modulo = [self.label_Error_Modulo_2,
           self.label_Error_Modulo_3, self.label_Error_Modulo_4, self.
           label_Error_Modulo_5, self.label_Error_Modulo_6]
1325     self.controles_limite_negativo_modulo = [self.
           label_limite_negativo_modulo1, self.
           label_limite_negativo_modulo2, self.
           label_limite_negativo_modulo3, self.
           label_limite_negativo_modulo4, self.

```

```

1326     label_limite_negativo_modulo5]
self.controles_limite_positivo_modulo = [self.
    label_limite_positivo_modulo1, self.
    label_limite_positivo_modulo2, self.
    label_limite_positivo_modulo3, self.
    label_limite_positivo_modulo4, self.
    label_limite_positivo_modulo5]
1327 self.controles_valores_error_angulo = [self.label_desfase_2, self.
    label_desfase_3, self.label_desfase_4, self.label_desfase_5,
    self.label_desfase_6]
1328 self.controles_limite_negativo_angulo = [self.
    label_limite_negativo_desfase1, self.
    label_limite_negativo_desfase2, self.
    label_limite_negativo_desfase3, self.
    label_limite_negativo_desfase4, self.
    label_limite_negativo_desfase5]
1329 self.controles_limite_positivo_angulo = [self.
    label_limite_positivo_desfase1, self.
    label_limite_positivo_desfase2, self.
    label_limite_positivo_desfase3, self.
    label_limite_positivo_desfase4, self.
    label_limite_positivo_desfase5]
1330
1331 def configurar_botones(self):
1332     """Funci n que configura la funcionalidad de los botones de la
    interfaz conect ndolo con el m todo que le corresponde.
    Tambi n configura el estilo en la interfaz.
1333
    Par metros:
1334     self — clase App
1335
1336     Atributos:
1337     self.pushButton_CambiarUsuario — bot n CAMBIAR USUARIO de la
    interfaz main.ui. Clase QPushButton de la librer a Qt5.
1338
1339     self.pushButton_prec — bot n PRECISION de la interfaz main.ui
    . Clase QPushButton de la librer a Qt5.
1340
1341     self.pushButton_DDPP — bot n DDPP de la interfaz main.ui.
    Clase QPushButton de la librer a Qt5.
1342
1343     self.pushButton_INICIO — bot n HV ON de la interfaz main.ui.
    Clase QPushButton de la librer a Qt5.
1344
1345     self.pushButton_PARAR — bot n HV OFF de la interfaz main.ui.
    Clase QPushButton de la librer a Qt5.
1346
1347     self.pushButton_GRABAR — bot n GRABAR de la interfaz main.ui.
    Clase QPushButton de la librer a Qt5.
1348
1349     self.pushButton_descartar — bot n DESCARTAR de la interfaz
    main.ui. Clase QPushButton de la librer a Qt5.
1350
1351     """
1352
1353     # Asignar acciones a los botones
1354     self.pushButton_CambiarUsuario.clicked.connect(self.
    cambiar_estado_log_in)
1355     self.pushButton_prec.clicked.connect(self.precision)

```

```

1356     self.pushButton_DDPP.clicked.connect(self.ddpp)
1357     self.pushButton_INICIO.clicked.connect(self.
1358         cambiar_estado_comenzar_ensayo)
1359     self.pushButton_PARAR.clicked.connect(self.cambiar_estado_parar)
1360     self.pushButton_GRABAR.clicked.connect(self.cambiar_estado_grabar)
1361     self.pushButton_descartar.clicked.connect(self.
1362         cambiar_estado_descartar)
1363
1364     # Estilo de los botones
1365     self.configurar_estilo_botones()
1366
1367     # Activar botones
1368     def activar_boton_cambiar_usuario(self):
1369         """Funci n que activa el bot n CAMBIAR USUARIO de la interfaz
1370             main.ui
1371
1372             Par metros:
1373                 self — clase App
1374
1375             Atributos:
1376                 self.pushButton_CambiarUsuario — bot n CAMBIAR USUARIO de la
1377                 interfaz main.ui. Clase QPushButton de la
1378                 librer a Qt5.
1379         """
1380         # Activar
1381         self.pushButton_CambiarUsuario.setEnabled(True)
1382
1383     def activar_boton_prec(self):
1384         """Funci n que activa el bot n PRECISION de la interfaz main.ui
1385
1386             Par metros:
1387                 self — clase App
1388
1389             Atributos:
1390                 self.pushButton_prec — bot n PRECISION de la interfaz main.ui.
1391                 Clase QPushButton de la librer a Qt5.
1392         """
1393         # Activar
1394         self.pushButton_prec.setEnabled(True)
1395
1396     def activar_boton_ddpp(self):
1397         """Funci n que activa el bot n DDPP de la interfaz main.ui
1398
1399             Par metros:
1400                 self — clase App
1401
1402             Atributos:
1403                 self.pushButton_DDPP — bot n DDPP de la interfaz main.ui.
1404                 Clase QPushButton de la librer a Qt5.
1405         """
1406         # Activar
1407         self.pushButton_DDPP.setEnabled(True)
1408
1409     def activar_boton_inicio(self):
1410         """Funci n que activa el bot n HV ON de la interfaz main.ui
1411
1412             Par metros:
1413                 self — clase App

```

```

1408
1409     Atributos:
1410         self.pushButton_INICIO — bot n HV ON de la interfaz main.ui.
1411             Clase QPushButton de la librer a Qt5.
1412     """
1413     # Activar
1414     self.pushButton_INICIO.setEnabled(True)
1415
1416 def activar_boton_parar(self):
1417     """Funci n que activa el bot n HV OFF de la interfaz main.ui
1418
1419     Par metros:
1420         self — clase App
1421
1422     Atributos:
1423         self.pushButton_PARAR — bot n HV OFF de la interfaz main.ui.
1424             Clase QPushButton de la librer a Qt5.
1425     """
1426     # Activar
1427     self.pushButton_PARAR.setEnabled(True)
1428
1429 def activar_boton_grabar(self):
1430     """Funci n que activa el bot n GRABAR de la interfaz main.ui
1431
1432     Par metros:
1433         self — clase App
1434
1435     Atributos:
1436         self.pushButton_GRABAR — bot n GRABAR de la interfaz main.ui.
1437             Clase QPushButton de la librer a Qt5.
1438     """
1439     # Activar
1440     self.pushButton_GRABAR.setEnabled(True)
1441
1442 def activar_boton_descartar(self):
1443     """Funci n que activa el bot n DESCARTAR de la interfaz main.ui
1444
1445     Par metros:
1446         self — clase App
1447
1448     Atributos:
1449         self.pushButton_descartar — bot n DESCARTAR de la interfaz
1450             main.ui. Clase QPushButton de la librer a Qt5.
1451     """
1452     # Activar
1453     self.pushButton_descartar.setEnabled(True)
1454
1455     # Desactivar botones
1456 def desactivar_boton_cambiar_usuario(self):
1457     """Funci n que desactiva el bot n CAMBIAR USUARIO de la interfaz
1458         main.ui
1459
1460     Par metros:
1461         self — clase App
1462
1463     Atributos:
1464         self.pushButton_CambiarUsuario — bot n CAMBIAR USUARIO de la
1465             interfaz main.ui. Clase QPushButton de la

```

```

1460         librer a Qt5.
1461         """
1462         # Desactivar
1463         self.pushButton_CambiarUsuario.setEnabled(False)
1464
1465     def desactivar_boton_prec(self):
1466         """Funci n que desactiva el bot n PRECISION de la interfaz main.
1467             ui
1468
1469         Par metros:
1470             self — clase App
1471
1472         Atributos:
1473             self.pushButton_prec — bot n PRECISION de la interfaz main.ui
1474             . Clase QPushButton de la librer a Qt5.
1475         """
1476         # Desactiva
1477         self.pushButton_prec.setEnabled(False)
1478
1479     def desactivar_boton_ddpp(self):
1480         """Funci n que activa el bot n DDPP de la interfaz main.ui
1481
1482         Par metros:
1483             self — clase App
1484
1485         Atributos:
1486             self.pushButton_DDPP — bot n DDPP de la interfaz main.ui.
1487             Clase QPushButton de la librer a Qt5.
1488         """
1489         # Desactivar
1490         self.pushButton_DDPP.setEnabled(False)
1491
1492     def desactivar_boton_inicio(self):
1493         """Funci n que desactiva el bot n HV ON de la interfaz main.ui
1494
1495         Par metros:
1496             self — clase App
1497
1498         Atributos:
1499             self.pushButton_INICIO — bot n HV ON de la interfaz main.ui.
1500             Clase QPushButton de la librer a Qt5.
1501         """
1502         # Desactiva
1503         self.pushButton_INICIO.setEnabled(False)
1504
1505     def desactivar_boton_parar(self):
1506         """Funci n que activa el bot n HV OFF de la interfaz main.ui
1507
1508         Par metros:
1509             self — clase App
1510
1511         Atributos:
1512             self.pushButton_PARAR — bot n HV OFF de la interfaz
1513             main.ui. Clase QPushButton de la librer a Qt5.
1514         """
1515         # Desactivar
1516         self.pushButton_PARAR.setEnabled(False)

```

```

1513 def desactivar_boton_grabar(self):
1514     """Funci n que desactiva el bot n GRABAR de la interfaz main.ui
1515
1516     Par metros:
1517         self — clase App
1518
1519     Atributos:
1520         self.pushButton_GRABAR — bot n GRABAR de la interfaz main.ui.
1521         Clase QPushButton de la librer a Qt5.
1522     """
1523     # Activar
1524     self.pushButton_GRABAR.setEnabled(False)
1525
1526 def desactivar_boton_descartar(self):
1527     """Funci n que desactiva el bot n DESCARTAR de la interfaz main.
1528     ui
1529
1530     Par metros:
1531         self — clase App
1532
1533     Atributos:
1534         self.pushButton_descartar — bot n DESCARTAR de la interfaz
1535         main.ui. Clase QPushButton de la librer a Qt5.
1536     """
1537     # Activar
1538     self.pushButton_descartar.setEnabled(False)
1539
1540 def interpretacion_valor_frecuencia(self):
1541     """Funci n que le asigna el valor adecuado a la fecuencia de
1542     ensayo dependiendo del valor de dis3 que se lea en
1543     el xml.
1544
1545     Valores posibles de dis3 y su correspondiente valor:
1546     f50 — 50
1547     f50_60 — 50
1548     f60 — 60
1549
1550     Parametros:
1551         self — clase App
1552
1553     Atributos:
1554         self.xml — clase ComunicacionWebService
1555         self.pz — clase PuneteZera
1556     """
1557     if self.xml.frecuencia == "f60":
1558         self.pz.frecuencia = "60"
1559     else:
1560         self.pz.frecuencia = "50"
1561
1562 # Hipotronics
1563 def definir_puntos_tension_hipotronics_ddpp(self):
1564     """Funci n que da valor a los puntos de medida del HIPOTRONICS en
1565     kV para ensayos de ddpp.
1566
1567     Parametros:
1568         self — clase App

```

```

1566
1567     Atributos
1568         self.h — clase Hipotronics
1569
1570         self.xml — clase ComunicacionWebService
1571
1572     Excepciones:
1573         Control de TypeError: si las casillas referentes a los puntos
1574             de medida est n vac as se le asignara un 0 a
1575             la tensi n.
1576
1577             """
1578 try:
1579     self.h.set_target_voltage1(self.xml.pto_1_ddpp)
1580     self.h.set_target_voltage2(self.xml.pto_2_ddpp)
1581     self.h.set_target_voltage3(self.xml.pto_3_ddpp)
1582     self.h.set_target_voltage4(self.xml.pto_4_ddpp)
1583     self.h.set_target_voltage5(self.xml.pto_5_ddpp)
1584 except:
1585     self.h.set_target_voltage1(0)
1586     self.h.set_target_voltage2(0)
1587     self.h.set_target_voltage3(0)
1588     self.h.set_target_voltage4(0)
1589     self.h.set_target_voltage5(0)
1590 def definir_puntos_tension_hipotronics_precision(self):
1591     """Funci n que da valor a los puntos de medida del HIPOTRONICS en
1592         kV. Los valores los recoge de la interfaz.
1593
1594     Se tiene en cuenta que Tension= (EVTp*punto(%))/(1.7*100000)
1595
1596     Parametros:
1597         self — clase App
1598
1599     Atributos:
1600         self.pz — clase PuenteZera
1601
1602         self.xml — clase ComunicacionWebService
1603
1604         self.h — clase Hipotronics
1605
1606         self.textEdit_L1 — textEdit de la interfaz main.ui que hace
1607             referencia a la casilla del primer punto de
1608             tensi n a medir en %. Clase QTextEdit de la librer a Qt5.
1609
1610         self.textEdit_L2 — textEdit de la interfaz main.ui que hace
1611             referencia a la casilla del segundo punto de
1612             tensi n a medir en %. Clase QTextEdit de la librer a Qt5.
1613
1614         self.textEdit_L3 — textEdit de la interfaz main.ui que hace
1615             referencia a la casilla del tercer punto de
1616             tensi n a medir en %. Clase QTextEdit de la librer a Qt5.

```

```

1617         self.textEdit_L5 — textEdit de la interfaz main.ui que hace
1618             referencia a la casilla del quinto punto de
1619             tensi n a medir en %. Clase QTextEdit de la librería Qt5.
1620
1621     Excepciones:
1622         Control de TypeError: si las casillas referentes a los puntos
1623             de medida est n vac as se le asignara un 0 a
1624             la tensi n.
1625
1626     """
1627     self.pz.EVTprim = self.xml.num_t_prim
1628     try:
1629         self.h.set_target_voltage1(float(self.pz.EVTprim) / 1.7 * float
1630             (self.textEdit_L1.toPlainText()) / 100000)
1631         self.h.set_target_voltage2(float(self.pz.EVTprim) / 1.7 * float
1632             (self.textEdit_L2.toPlainText()) / 100000)
1633         self.h.set_target_voltage3(float(self.pz.EVTprim) / 1.7 * float
1634             (self.textEdit_L3.toPlainText()) / 100000)
1635         self.h.set_target_voltage4(float(self.pz.EVTprim) / 1.7 * float
1636             (self.textEdit_L4.toPlainText()) / 100000)
1637         self.h.set_target_voltage5(float(self.pz.EVTprim) / 1.7 * float
1638             (self.textEdit_L5.toPlainText()) / 100000)
1639     except:
1640         self.h.set_target_voltage1(0)
1641         self.h.set_target_voltage2(0)
1642         self.h.set_target_voltage3(0)
1643         self.h.set_target_voltage4(0)
1644         self.h.set_target_voltage5(0)
1645
1646     self.h.set_target_voltage1(round(self.h.targetVoltage1, 2))
1647     self.h.set_target_voltage2(round(self.h.targetVoltage2, 2))
1648     self.h.set_target_voltage3(round(self.h.targetVoltage3, 2))
1649     self.h.set_target_voltage4(round(self.h.targetVoltage4, 2))
1650     self.h.set_target_voltage5(round(self.h.targetVoltage5, 2))
1651
1652     def definir_datos_hipotronics(self):
1653         """Funci n que da valor al Trip Voltage y los datos de la rampa
1654             del Hipotronics: inclinaci n y tiempo. Los valores
1655             los recoge de la interfaz
1656
1657         Par metros:
1658             self — clase App
1659
1660         Atributos:
1661             self.h — clase Hipotronics
1662
1663             self.textEdit_Rampa1 — textEdit de la interfaz main.ui que
1664                 hace referencia a la casilla del valor de la
1665                 inclinaci n de la rampa del Hipotronics del primer punto.
1666                 Clase QTextEdit de la librería Qt5.
1667
1668             self.textEdit_Tiempo1 — textEdit de la interfaz main.ui que
1669                 hace referencia a la casilla del tiempo de
1670                 espera de la rampa del Hipotronics en el primer punto. Clase
1671                 QTextEdit de la librería Qt5.
1672
1673             self.textEdit_Tiempo2 — textEdit de la interfaz main.ui que
1674                 hace referencia a la casilla del tiempo de

```



```

1662     espera de la rampa del Hipotronics en el segundo punto. Clase
1663         QTextEdit de la librer a Qt5.
1664
1665     self.textEdit_Tiempo3 — textEdit de la interfaz main.ui que
1666         hace referencia a la casilla del tiempo de
1667     espera de la rampa del Hipotronics en el tercer punto. Clase
1668         QTextEdit de la librer a Qt5.
1669
1670     self.textEdit_Tiempo4 — textEdit de la interfaz main.ui que
1671         hace referencia a la casilla del tiempo de
1672     espera de la rampa del Hipotronics en el cuarto punto. Clase
1673         QTextEdit de la librer a Qt5.
1674
1675     self.textEdit_Tiempo5 — textEdit de la interfaz main.ui que
1676         hace referencia a la casilla del tiempo de
1677     espera de la rampa del Hipotronics en el quinto punto. Clase
1678         QTextEdit de la librer a Qt5.
1679
1680     """
1681
1682     self.h.set_ramp1(self.textEdit_Rampa1.toPlainText())
1683
1684     self.h.set_dwell1(self.textEdit_Tiempo1.toPlainText())
1685     self.h.dwell2 = self.textEdit_Tiempo2.toPlainText()
1686     self.h.dwell3 = self.textEdit_Tiempo3.toPlainText()
1687     self.h.dwell4 = self.textEdit_Tiempo2.toPlainText()
1688     self.h.dwell5 = self.textEdit_Tiempo3.toPlainText()
1689
1690     self.h.tripVoltage = (float(self.h.targetVoltage1) + 1).__str__()
1691
1692 # Funciones de comprobaci n
1693 def comprobacion_si_ensayado_precision(self):
1694     """Funci n que comprueba si ese aparato ya ha sido ensayada y
1695         tiene valores guardados en SAIO para el ensayo de
1696         precisi n.
1697
1698     1. Se inicializa el atributo ensayado, indicando que en un
1699         principio ese aparato no ha sido ensayado.
1700     2. Comprueba si el error en m dulo del primer punto para ese
1701         aparato tiene alg n valor guardado.
1702     En caso de tener:
1703     3. Comprueba si todos los valores del error en m dulo est n
1704         dentro de los l mites
1705     4. Comprueba si todos los valores del error en ngulo est n
1706         dentro de los l mites
1707     5. Se indica en el atributo ensayado que el aparato s que
1708         est ensayado.7
1709     6. Se comprueba el atributo ensayado. En caso de s estar ensayado
1710         :
1711     7. Se llama a la funci n de nueva tanda
1712     En caso de no estar ensayado:
1713     7. Se activa el bot n HV ON
1714
1715     Par metros:
1716     self — clase App
1717
1718     Atributos:
1719     ensayado — boolean que nos indica si el aparato est ensayado
1720
1721     .

```

```

1705         True —> Ensayado
1706         False —> No ensayado
1707
1708     self.xml — clase ComunicaionWebService
1709
1710     self.controles_valores_error_modulo — lista que hace
1711         referencia a los errores en modulo para ensayo de
1712         precisi n
1713
1714     self.controles_valores_error_angulo — lista que hace
1715         referencia a los errores en ngulo para ensayo de
1716         precisi n
1717
1718     """
1719     # si el valor de alguno de los errores de modulo o angulo no es
1720     # blanco considerar aparato tiene hecho el
1721     # ensayo de precision. Avisar. Incialmente mostrarndo mensaje.
1722     # Posteriormente preguntar si se quiere hacer una
1723     # tanda nueva
1724     ensayado = False
1725     if self.xml.errorModulo1 is not None:
1726         for i in range(len(self.controles_valores_error_modulo)):
1727             self.comprobar_error_modulo(self.
1728                 controles_valores_error_modulo[i].text(), i)
1729             self.comprobar_error_angulo(self.
1730                 controles_valores_error_angulo[i].text(), i)
1731         ensayado = True
1732     if ensayado:
1733         self.nueva_tanda()
1734     else:
1735         self.activar_boton_inicio()
1736
1737 def comprobacion_si_ensayado_ddpp(self):
1738     """Funci n que comprueba si ese aparato ya ha sido ensayada y
1739     tiene valores guardados en SAIO para el ensayo de
1740     descargas parciales.
1741
1742     1. Se inicializa el atributo ensayado, indicando que en un
1743     principio ese aparato no ha sido ensayado.
1744     2. Comprueba si el error en modulo del primer punto para ese
1745     aparato tiene alg n valor guardado.
1746     En caso de tener:
1747     3. Se indica en el atributo ensayado que el aparato s que
1748     est ensayado.7
1749     4. Se comprueba el atributo ensayado. En caso de s estar ensayado
1750     :
1751     5. Se llama a la funci n de nueva tanda
1752     En caso de no estar ensayado:
1753     5. Se activa el bot n HV ON
1754
1755     Par metros:
1756     self — clase App
1757
1758     Atributos:
1759     ensayado — boolean que nos indica si el aparato est ensayado
1760     .
1761     True —> Ensayado
1762     False —> No ensayado

```

```

1751         self.xml — clase ComunicaionWebService
1752
1753     """
1754     # si el valor de alfuno de los valores ddpp no es blanco considerar
1755     # que el aparato tiene hecho el ensayo ddpp.
1756     ensayado = False
1757     if self.xml.valor_ddpp_1 is not None:
1758         ensayado = True
1759     if ensayado:
1760         self.nueva_tanda()
1761     else:
1762         self.activar_boton_inicio()
1763
1764 def comprobar_tension(self, tension_a_medir, punto, horquilla):
1765     """Funci n que comprueba si la tensi n que se est midiendo es
1766     la del punto a medir, con una horquilla de margen.
1767
1768     1. Se le da el valor del punto a medir al punto actual.
1769     2. Se hace una lectura de las medidad el ZERA
1770     3. Si el punto actual es el punto que queremos medir:
1771         4. Si la medida de la tensi n que leemos del Zera es la
1772            tensi n que queremos medir, con la horquilla de margen:
1773             5. Se vuelve a hacer una lectura de las medidas del Zera
1774             6. Se escriben las medidas obtenidas de la tensi n , el
1775                erron en m dulo y el desfase en la interfaz.
1776             7. Se comprueba que los errores en m dulo y ngulo est n
1777                dentro de los l mites
1778             8. Se actualiza el punto actual pasando al siguiente
1779     4. Si la medida de la tensi n que leemos o la tensi n a medir
1780        es menor que 0.8:
1781         5. Se escribe "-" en la interfaz.
1782         6. Se actualiza el punto actual pasando al siguiente
1783     4. En cualquier otro caso:
1784         5. Se vuelve a llamar a la funci n comprobar_tension
1785
1786 Par metros:
1787     self — clase App
1788
1789     tension_a_medir — tensi n a la que se quiere guardar el valor
1790     del ensayo
1791
1792     punto — valor que indica el punto de ensayo referente a esa
1793     tensi n a medir
1794
1795     horquilla — margen aceptable de la medida de tensi n
1796
1797 Atributos:
1798     punto_actual — punto en el que se encuentra el ensayo
1799
1800     self.pz — clase PuenteZera
1801
1802     elf.controles_lecturas_precision — lista que hace referencia a
1803     los datos de las medidas de precision
1804
1805     self.controles_valores_error_modulo — lista que hace
1806     referencia a los errores en m dulo para ensayo de
1807     precisi n

```

```

1799
1800     self. controles_valores_error_angulo — lista que hace
1801     referencia a los errores en ngulo para ensayo de
1802     precisi n
1803
1804     """
1805     punto_actual = punto
1806     self.pz.leer()
1807     if punto_actual == punto:
1808         if float(tension_a_medir) - float(horquilla) < float(self.pz.
1809             LPN).__round__(2) < float(
1810                 tension_a_medir) + float(horquilla):
1811             # TODO COMPROBAR
1812             self.pz.leer()
1813             self.controles_lecturas_precision[punto].setText(self.pz.
1814                 LPN)
1815             self.controles_valores_error_modulo[punto].setText(self.pz.
1816                 ErrorModulo)
1817             self.comprobar_error_modulo(self.pz.ErrorModulo, punto)
1818             self.controles_valores_error_angulo[punto].setText(self.pz.
1819                 ErrorAngulo)
1820             self.comprobar_error_angulo(self.pz.ErrorAngulo, punto)
1821             punto_actual += 1
1822         elif float(tension_a_medir) < 0.8 or float(self.pz.LPN) < 0.8:
1823             self.controles_lecturas_precision[punto].setText("-")
1824             self.controles_valores_error_modulo[punto].setText("-")
1825             self.controles_valores_error_angulo[punto].setText("-")
1826             punto_actual += 1
1827         else:
1828             self.comprobar_tension(tension_a_medir, punto, horquilla)
1829
1830     def comprobar_error_modulo(self, error, punto):
1831         """Funci n que comprueba si el error en m dulo en un punto est
1832         dentro de los errores establecidos.
1833         Si no lo est pone su casilla correspondiente en la interfaz en
1834         rojo.
1835
1836         Par metros:
1837         self — clase App
1838
1839         error — valor del error en m dulo que se quiere comprobar
1840
1841         punto — el punto de ensayo correspondiente a ese error
1842
1843         Atributos:
1844         self.controles_limite_negativo_modulo — lista que hace
1845         referencia a los limites negativos del error en
1846         m dulo para ensayo de precisi n
1847
1848         self.controles_limite_positivo_modulo — lista que hace
1849         referencia a los limites positivos del error en
1850         m dulo para ensayo de precisi n
1851
1852         self.controles_valores_error_modulo — lista que hace
1853         referencia a los errores en m dulo para ensayo de
1854         precisi n
1855
1856         """

```

```

1847     if not float(self.controles_limite_negativo_modulo[punto].text()) <
1848         float(error) < float(self.controles_limite_positivo_modulo[
1849             punto].text()):
1850         self.fondo_rojo(self.controles_valores_error_modulo[punto])
1851
1852     def comprobar_error_angulo(self, error, punto):
1853         """Funci n que comprueba si el error en ngulo en un punto est
1854             dentro de los errores establecidos.
1855             Si no lo est pone su casilla correspondiente en la interfaz en
1856             rojo.
1857
1858             Par metros:
1859             self — clase App
1860
1861             error — valor del error en ngulo que se quiere comprobar
1862
1863             punto — el punto de ensayo correspondiente a ese error
1864
1865             Atributos:
1866             self.controles_limite_negativo_angulo — lista que hace
1867                 referencia a los limites negativos del error en
1868                 ngulo para ensayo de precisi n
1869
1870             self.controles_limite_positivo_angulo — lista que hace
1871                 referencia a los limites positivos del error en
1872                 ngulo para ensayo de precisi n
1873
1874             self.controles_valores_error_angulo — lista que hace
1875                 referencia a los errores en ngulo para ensayo de
1876                 precisi n
1877
1878             """
1879     if not float(self.controles_limite_negativo_angulo[punto].text()) <
1880         float(error) < float(self.controles_limite_positivo_angulo[
1881             punto].text()):
1882         self.fondo_rojo(self.controles_valores_error_angulo[punto])
1883
1884 # INTERFAZ
1885 # LOGO
1886 def add_logo(self):
1887     """M todo para aadir el logo de la empresa a la interfaz main.ui
1888     .
1889     Logo.jpg se encuentra en la carpeta IMG.
1890
1891     Parametros:
1892     self — clase App
1893
1894     Atributos:
1895     pixmap — clase QPixmap de la librer a Qt5
1896
1897     self.label_logo — label de la interfaz main.ui que hace
1898         referencia al logo corporativo.
1899     Clase QLabel de la librer a Qt5.
1900     """
1901     pixmap = QPixmap('../src/IMG/Logo.jpg')
1902     self.label_logo.setPixmap(pixmap)
1903     self.resize(pixmap.width(), pixmap.height())

```

```

1894
1895 # BOTONES
1896 def configurar_estilo_botones(self):
1897     """Funci n que configuro el estilo de los botones en la interfaz
1898         main.ui.
1899
1900     Par metros:
1901         self — clase App
1902
1903     Atributos:
1904         self.pushButton_INICIO — bot n HV ON de la interfaz main.ui.
1905             Clase QPushButton de la librer a Qt5.
1906
1907         self.pushButton_PARAR — bot n HV OFF de la interfaz main.ui.
1908             Clase QPushButton de la librer a Qt5.
1909
1910         self.pushButton_GRABAR — bot n GRABAR de la interfaz main.ui.
1911             Clase QPushButton de la librer a Qt5.
1912
1913         self.pushButton_descartar — bot n DESCARTAR de la interfaz
1914             main.ui. Clase QPushButton de la librer a Qt5.
1915
1916     """
1917     # Botones
1918     self.fondo_verde_arteche(self.pushButton_INICIO)
1919     self.fondo_verde_arteche(self.pushButton_GRABAR)
1920     self.fondo_rojo(self.pushButton_PARAR)
1921     self.fondo_rojo(self.pushButton_descartar)
1922
1923 # Refrescos de pantalla
1924 def refrescar_pantalla_log_in(self):
1925     """Funci n que refresca la casilla que indica el usuario en la
1926         interfaz main.ui.
1927
1928     Par metros:
1929         self — clase App
1930
1931     Atributos:
1932         self.label_usuario — label de la interfaz main.ui que hace
1933             referencia al usuario del log in.
1934             Clase QLabel de la librer a Qt5.
1935
1936         self.usuario — usuario para el log in
1937
1938     """
1939     # mostrar el usuario
1940     self.label_usuario.setStyleSheet(u"font: 75 20pt\ "Ms Gotham Book\ "
1941         ;\n")
1942     self.label_usuario.setText(self.usuario)
1943
1944 def inicializar_datos_entrada(self):
1945     """Funci n que limpia las casillas de los datos de entrada de la
1946         interfaz main.ui. Pone en modo editable la
1947         casilla de la OF.
1948
1949     Par metros:
1950         self — clase App

```

```

1943     Atributos:
1944         self.controles_datos_entrada — lista que hace referencia a los
1945             datos de entrada
1946
1947         self.textEdit_OFS — textEdit de la interfaz main.ui que hace
1948             referencia a la casilla de OF.
1949         Clase QTextEdit de la librer a Qt5.
1950
1951     """
1952     for c in self.controles_datos_entrada:
1953         self.fondo_azul_arteche(c)
1954         self.texto_a_blanco(c)
1955     self.editable(self.textEdit_OFS)
1956
1957     # TODO ense ar a unai
1958     def mostrar_info_general(self):
1959         """Funci n que actualiza el apartado de Informaci n General de la
1960             interfaz main.ui.
1961
1962         Parametros:
1963         self — clase App
1964
1965         Atributos:
1966         self.label_columna_titulos1 — label de la interfaz main.ui que
1967             hace referencia a la primera columna de
1968             t tulos de la informaci n general. Clase QLabel de la
1969             librer a Qt5.
1970
1971         self.label_columna_titulos2 — label de la interfaz main.ui que
1972             hace referencia a la segunda columna de
1973             t tulos de la informaci n general. Clase QLabel de la
1974             librer a Qt5.
1975
1976         self.label_columna_valores1 — label de la interfaz main.ui que
1977             hace referencia a la primera columna de
1978             valores de la informaci n general. Clase QLabel de la
1979             librer a Qt5.
1980
1981         self.label_columna_valores2 — label de la interfaz main.ui que
1982             hace referencia a la segunda columna de
1983             valores de la informaci n general. Clase QLabel de la
1984             librer a Qt5.
1985
1986         self.xml — clase ComunicacionWebService
1987
1988     """
1989     self.fondo_azul_arteche(self.label_columna_titulos1)
1990     self.fondo_azul_arteche(self.label_columna_titulos2)
1991     self.label_columna_titulos1.setText("CLIENTE:\nO.F.:\nO.V.:\nModelo
1992         :\nTemperatura:")
1993     self.label_columna_titulos2.setText("Clase:\nFactor V:\nNorma:\n
1994         Carga C:\nCarga R:")
1995
1996     self.label_columna_valores1.setStyleSheet(
1997         u"background-color: rgb(197, 216, 130);\n"font: 75 14pt\Ms
1998             Gotham Bold\";\n")
1999     self.label_columna_valores2.setStyleSheet(

```

```

1986         u"background-color: rgb(197, 216, 130);\n"font: 75 14pt\"Ms
1987             Gotham Bold\";\n")
1988     self.label_columna_valores1.setText(self.xml.nombre_cliente + "\n"
1989         + self.xml.orden_fab + "\n" +
1990         self.xml.OV_IN_OUT + "\n" +
1991         self.xml.modelo + "\n" +
1992         self.xml.temp_min +
1993         '/' + self.xml.temp_max)
1994
1995     self.label_columna_valores2.setText(
1996         self.xml.CLASE + "\n" + self.xml.factor_tension + "\n" + self.
1997         xml.norma + "\n" + self.xml.carga_capac +
1998         "\n" + self.xml.carga_resist)
1999
2000     def mostrar_info_precision(self):
2001         """Funci n que muestra la informaci n referente al ensayo de
2002             precisi n y configura el estilo de botones y casillas
2003             en la interfaz main.ui. Limpia lo referente al ensayo de descargas
2004             parciales.
2005
2006             Par metros:
2007                 self — clase App
2008
2009             Atributos:
2010                 self.textEdit_horquilla — textEdit de la interfaz main.ui que
2011                 hace referencia a la casilla de la
2012                 horquilla. Clase QTextEdit de la librer a Qt5.
2013
2014                 self.pushButton_prec — bot n PRECISION de la interfaz main.ui
2015                 . Clase QPushButton de la librer a Qt5.
2016
2017                 self.pushButton_DDPP — bot n DDPP de la interfaz main.ui.
2018                 Clase QPushButton de la librer a Qt5.
2019
2020                 self.textEdit_OFS — textEdit de la interfaz main.ui que hace
2021                 referencia a la casilla de OF.
2022                 Clase QTextEdit de la librer a Qt5.
2023
2024                 self.controles_lecturas_zera — lista que hace referencia a los
2025                 datos de las lecturas del puente Zera
2026
2027                 self.controles_lecturas_precision — lista que hace referencia
2028                 a los datos de las medidas de precision
2029
2030                 self.controles_valores_error_modulo — lista que hace
2031                 referencia a los errores en m dulo para ensayo de
2032                 precisi n
2033
2034                 self.controles_valores_error_angulo — lista que hace
2035                 referencia a los errores en ngulo para ensayo de
2036                 precisi n
2037
2038                 self.textEdit_L1 — textEdit de la interfaz main.ui que hace
2039                 referencia a la casilla del primer punto de
2040                 tensi n a medir en %. Clase QTextEdit de la librer a Qt5.
2041
2042 """

```



2027 *self.textEdit\_L2* — *textEdit* de la interfaz *main.ui* que hace  
referencia a la casilla del segundo punto de  
2028 *tensi n* a medir en %. Clase *QTextEdit* de la librería *Qt5*.  
2029  
2030 *self.textEdit\_L3* — *textEdit* de la interfaz *main.ui* que hace  
referencia a la casilla del tercer punto de  
2031 *tensi n* a medir en %. Clase *QTextEdit* de la librería *Qt5*.  
2032  
2033 *self.textEdit\_L4* — *textEdit* de la interfaz *main.ui* que hace  
referencia a la casilla del cuarto punto de  
2034 *tensi n* a medir en %. Clase *QTextEdit* de la librería *Qt5*.  
2035  
2036 *self.textEdit\_L5* — *textEdit* de la interfaz *main.ui* que hace  
referencia a la casilla del quinto punto de  
2037 *tensi n* a medir en %. Clase *QTextEdit* de la librería *Qt5*.  
2038  
2039 *self.label\_Error\_Modulo\_2* — *label* de la interfaz *main.ui* que  
hace referencia al valor del error en *m dulo*  
2040 medido en el primer punto del ensayo de *precisi n* en %. Clase  
*QLabel* de la librería *Qt5*.  
2041  
2042 *self.label\_Error\_Modulo\_3* — *label* de la interfaz *main.ui* que  
hace referencia al valor del error en *m dulo*  
2043 medido en el segundo punto del ensayo de *precisi n* en %. Clase  
*QLabel* de la librería *Qt5*.  
2044  
2045 *self.label\_Error\_Modulo\_4* — *label* de la interfaz *main.ui* que  
hace referencia al valor del error en *m dulo*  
2046 medido en el tercer punto del ensayo de *precisi n* en %. Clase  
*QLabel* de la librería *Qt5*.  
2047  
2048 *self.label\_Error\_Modulo\_5* — *label* de la interfaz *main.ui* que  
hace referencia al valor del error en *m dulo*  
2049 medido en el cuarto punto del ensayo de *precisi n* en %. Clase  
*QLabel* de la librería *Qt5*.  
2050  
2051 *self.label\_Error\_Modulo\_6* — *label* de la interfaz *main.ui* que  
hace referencia al valor del error en *m dulo*  
2052 medido en el quinto punto del ensayo de *precisi n* en %. Clase  
*QLabel* de la librería *Qt5*.  
2053  
2054 *self.label\_desfase\_2* — *label* de la interfaz *main.ui* que hace  
referencia al valor del desfase medido en  
2055 el primer punto del ensayo de *precisi n* en %. Clase *QLabel* de  
la librería *Qt5*.  
2056  
2057 *self.label\_desfase\_3* — *label* de la interfaz *main.ui* que hace  
referencia al valor del desfase medido en  
2058 el segundo punto del ensayo de *precisi n* en %. Clase *QLabel* de  
la librería *Qt5*.  
2059  
2060 *self.label\_desfase\_4* — *label* de la interfaz *main.ui* que hace  
referencia al valor del desfase medido en  
2061 el tercer punto del ensayo de *precisi n* en %. Clase *QLabel* de  
la librería *Qt5*.  
2062  
2063 *self.label\_desfase\_5* — *label* de la interfaz *main.ui* que hace  
referencia al valor del desfase medido en

2064 *el cuarto punto del ensayo de precisión en %. Clase QLabel de*  
*la librería Qt5.*

2065

2066 *self.label\_desfase\_6 — label de la interfaz main.ui que hace*  
*referencia al valor del desfase medido en*

2067 *el quinto punto del ensayo de precisión en %. Clase QLabel de*  
*la librería Qt5.*

2068

2069 *self.label\_tanda\_prec — label de la interfaz main.ui que hace*  
*referencia a la tanda de ensayos del ensayo*

2070 *de precisión. Clase QLabel de la librería Qt5.*

2071

2072 *self.label\_tanda\_ddpp — label de la interfaz main.ui que hace*  
*referencia a la tanda de ensayos del ensayo*

2073 *de descargas parciales. Clase QLabel de la librería Qt5.*

2074

2075 *self.controles\_limite\_negativo\_angulo — lista que hace*  
*referencia a los límites negativos del error en*

2076 *ángulo para ensayo de precisión*

2077

2078 *self.controles\_limite\_negativo\_modulo — lista que hace*  
*referencia a los límites negativos del error en*

2079 *módulo para ensayo de precisión*

2080

2081 *self.controles\_limite\_positivo\_angulo — lista que hace*  
*referencia a los límites positivos del error en*

2082 *ángulo para ensayo de precisión*

2083

2084 *self.controles\_limite\_positivo\_modulo — lista que hace*  
*referencia a los límites positivos del error en*

2085 *módulo para ensayo de precisión*

2086

2087 *self.label\_limite\_negativo\_modulo1 — label de la interfaz main*  
*.ui que hace referencia al límite negativo*

2088 *del error en módulo del ensayo de descargas parciales en el*  
*primer punto. Clase QLabel de la librería Qt5.*

2089

2090 *self.label\_limite\_negativo\_modulo2 — label de la interfaz main*  
*.ui que hace referencia al límite negativo*

2091 *del error en módulo del ensayo de descargas parciales en el*  
*segundo punto. Clase QLabel de la librería Qt5.*

2092

2093 *self.label\_limite\_negativo\_modulo3 — label de la interfaz main*  
*.ui que hace referencia al límite negativo*

2094 *del error en módulo del ensayo de descargas parciales en el*  
*tercer punto. Clase QLabel de la librería Qt5.*

2095

2096 *self.label\_limite\_negativo\_modulo4 — label de la interfaz main*  
*.ui que hace referencia al límite negativo*

2097 *del error en módulo del ensayo de descargas parciales en el*  
*cuarto punto. Clase QLabel de la librería Qt5.*

2098

2099 *self.label\_limite\_negativo\_modulo5 — label de la interfaz main*  
*.ui que hace referencia al límite negativo*

2100 *del error en módulo del ensayo de descargas parciales en el*  
*quinto punto. Clase QLabel de la librería Qt5.*

2101

2102 *self.label\_limite\_positivo\_modulo1* — label de la interfaz main  
2103 *.ui* que hace referencia al limite positivo  
del error en modulo del ensayo de descargas parciales en el  
2104 primer punto. Clase *QLabel* de la libreria *Qt5*.  
2105  
2106 *self.label\_limite\_positivo\_modulo2* — label de la interfaz main  
*.ui* que hace referencia al limite positivo  
del error en modulo del ensayo de descargas parciales en el  
2107 segundo punto. Clase *QLabel* de la libreria *Qt5*.  
2108  
2109 *self.label\_limite\_positivo\_modulo3* — label de la interfaz main  
*.ui* que hace referencia al limite positivo  
del error en modulo del ensayo de descargas parciales en el  
2110 tercer punto. Clase *QLabel* de la libreria *Qt5*.  
2111  
2112 *self.label\_limite\_positivo\_modulo4* — label de la interfaz main  
*.ui* que hace referencia al limite positivo  
del error en modulo del ensayo de descargas parciales en el  
2113 cuarto punto. Clase *QLabel* de la libreria *Qt5*.  
2114  
2115 *self.label\_limite\_positivo\_modulo5* — label de la interfaz main  
*.ui* que hace referencia al limite positivo  
del error en modulo del ensayo de descargas parciales en el  
2116 quinto punto. Clase *QLabel* de la libreria *Qt5*.  
2117  
2118 *self.label\_limite\_negativo\_desfase1* — label de la interfaz  
*main.ui* que hace referencia al limite negativo  
del desfase del ensayo de descargas parciales en el primer  
2119 punto. Clase *QLabel* de la libreria *Qt5*.  
2120  
2121 *self.label\_limite\_negativo\_desfase2* — label de la interfaz  
*main.ui* que hace referencia al limite negativo  
del desfase del ensayo de descargas parciales en el segundo  
2122 punto. Clase *QLabel* de la libreria *Qt5*.  
2123  
2124 *self.label\_limite\_negativo\_desfase3* — label de la interfaz  
*main.ui* que hace referencia al limite negativo  
del desfase del ensayo de descargas parciales en el tercer  
2125 punto. Clase *QLabel* de la libreria *Qt5*.  
2126  
2127 *self.label\_limite\_negativo\_desfase4* — label de la interfaz  
*main.ui* que hace referencia al limite negativo  
del desfase del ensayo de descargas parciales en el cuarto  
2128 punto. Clase *QLabel* de la libreria *Qt5*.  
2129  
2130 *self.label\_limite\_negativo\_desfase5* — label de la interfaz  
*main.ui* que hace referencia al limite negativo  
del desfase del ensayo de descargas parciales en el quinto  
2131 punto. Clase *QLabel* de la libreria *Qt5*.  
2132  
2133 *self.label\_limite\_positivodesfase\_desfase1* — label de la  
interfaz *main.ui* que hace referencia al limite positivo  
del desfase del ensayo de descargas parciales en el primer  
2134 punto. Clase *QLabel* de la libreria *Qt5*.  
2135  
2136 *self.label\_limite\_positivo\_desfase2* — label de la interfaz  
*main.ui* que hace referencia al limite positivo

```

2136         del desfase del ensayo de descargas parciales en el segundo
           punto. Clase QLabel de la librería Qt5.
2137
2138         self.label_limite_positivo_desfase3 — label de la interfaz
           main.ui que hace referencia al limite positivo
2139         del desfase del ensayo de descargas parciales en el tercer
           punto. Clase QLabel de la librería Qt5.
2140
2141         self.label_limite_positivo_desfase4 — label de la interfaz
           main.ui que hace referencia al limite positivo
2142         del desfase del ensayo de descargas parciales en el cuarto
           punto. Clase QLabel de la librería Qt5.
2143
2144         self.label_limite_positivo_desfase5 — label de la interfaz
           main.ui que hace referencia al limite positivo
2145         del desfase del ensayo de descargas parciales en el quinto
           punto. Clase QLabel de la librería Qt5.
2146
2147         self.controles_configuracion_zera — lista que hace referencia
           a los datos de la configuración del puente Zera
2148
2149         self.textEdit_Np — textEdit de la interfaz main.ui que hace
           referencia a la casilla de la tensión
2150         primaria del patrón. Clase QTextEdit de la librería Qt5.
2151
2152         self.textEdit_Ns — textEdit de la interfaz main.ui que hace
           referencia a la casilla de la tensión
2153         secundaria del patrón. Clase QTextEdit de la librería Qt5.
2154
2155         self.textEdit_EVTp — textEdit de la interfaz main.ui que hace
           referencia a la casilla de la tensión
2156         primaria de la tensión a medir. Clase QTextEdit de la librería
           Qt5.
2157
2158         self.textEdit_EVTs — textEdit de la interfaz main.ui que hace
           referencia a la casilla de la tensión
2159         secundaria de la tensión a medir. Clase QTextEdit de la
           librería Qt5.
2160
2161         self.textEdit_freq — textEdit de la interfaz main.ui que hace
           referencia a la casilla de la frecuencia.
2162         Clase QTextEdit de la librería Qt5.
2163
2164         self.xml — clase ComunicacionWebService
2165
2166         self.pz — clase PuenteZera
2167         """
2168         self.textEdit_horquilla.setText(HORQUILLA)
2169         self.fondo_blanco(self.pushButton_DDPP)
2170         self.fondo_azul_arteche(self.pushButton_prec)
2171         self.no_editable(self.textEdit_OFS)
2172
2173         for c in self.controles_lecturas_zera:
2174             self.fondo_azul_arteche(c)
2175         for c in self.controles_lecturas_precision:
2176             self.fondo_azul_arteche(c)
2177         for c in self.controles_valores_error_modulo:
2178             self.fondo_azul_arteche(c)

```

```

2179     for c in self.controles_valores_error_angulo:
2180         self.fondo_azul_arteche(c)
2181
2182     self.textEdit_L1.setText(self.xml.nivel_1_precision)
2183     self.textEdit_L2.setText(self.xml.nivel_2_precision)
2184     self.textEdit_L3.setText(self.xml.nivel_3_precision)
2185     self.textEdit_L4.setText(self.xml.nivel_4_precision)
2186     self.textEdit_L5.setText(self.xml.nivel_5_precision)
2187
2188     self.label_Error_Modulo_2.setText(self.xml.errorModulo1)
2189     self.label_Error_Modulo_3.setText(self.xml.errorModulo2)
2190     self.label_Error_Modulo_4.setText(self.xml.errorModulo3)
2191     self.label_Error_Modulo_5.setText(self.xml.errorModulo4)
2192     self.label_Error_Modulo_6.setText(self.xml.errorModulo5)
2193
2194     self.label_desfase_2.setText(self.xml.errorAngulo1)
2195     self.label_desfase_3.setText(self.xml.errorAngulo2)
2196     self.label_desfase_4.setText(self.xml.errorAngulo3)
2197     self.label_desfase_5.setText(self.xml.errorAngulo4)
2198     self.label_desfase_6.setText(self.xml.errorAngulo5)
2199
2200     # Info tanda
2201     self.fondo_naranja(self.label_tanda_prec)
2202     self.label_tanda_prec.setText(self.xml.NUM_TANDA + " " + self.xml
2203         .DESCRIPCION_TANDA)
2204
2205     self.fondo_gris(self.label_tanda_ddpp)
2206     self.texto_a_blanco(self.label_tanda_ddpp)
2207
2208     # Limites precision
2209     for c in self.controles_limite_negativo_modulo:
2210         self.fondo_verde_arteche(c)
2211     for c in self.controles_limite_positivo_modulo:
2212         self.fondo_verde_arteche(c)
2213     for c in self.controles_limite_negativo_angulo:
2214         self.fondo_verde_arteche(c)
2215     for c in self.controles_limite_positivo_angulo:
2216         self.fondo_verde_arteche(c)
2217
2218     self.label_limite_negativo_modulo1.setText(self.xml.
2219         lim_modulo_negativo_1)
2220     self.label_limite_negativo_modulo2.setText(self.xml.
2221         lim_modulo_negativo_2)
2222     self.label_limite_negativo_modulo3.setText(self.xml.
2223         lim_modulo_negativo_3)
2224     self.label_limite_negativo_modulo4.setText(self.xml.
2225         lim_modulo_negativo_4)
2226     self.label_limite_negativo_modulo5.setText(self.xml.
2227         lim_modulo_negativo_5)
2228
2229     self.label_limite_positivo_modulo1.setText(self.xml.
2230         lim_modulo_positivo_1)
2231     self.label_limite_positivo_modulo2.setText(self.xml.
2232         lim_modulo_positivo_2)
2233     self.label_limite_positivo_modulo3.setText(self.xml.
2234         lim_modulo_positivo_3)
2235     self.label_limite_positivo_modulo4.setText(self.xml.
2236         lim_modulo_positivo_4)

```

```

2227     self.label_limite_positivo_modulo5.setText(self.xml.
2228         lim_modulo_positivo_5)
2229
2230     self.label_limite_negativo_desfase1.setText(self.xml.
2231         lim_angulo_1_negativo)
2232     self.label_limite_negativo_desfase2.setText(self.xml.
2233         lim_angulo_2_negativo)
2234     self.label_limite_negativo_desfase3.setText(self.xml.
2235         lim_angulo_3_negativo)
2236     self.label_limite_negativo_desfase4.setText(self.xml.
2237         lim_angulo_4_negativo)
2238     self.label_limite_negativo_desfase5.setText(self.xml.
2239         lim_angulo_5_negativo)
2240
2241     self.label_limite_positivo_desfase1.setText(self.xml.
2242         lim_angulo_1_positivo)
2243     self.label_limite_positivo_desfase2.setText(self.xml.
2244         lim_angulo_2_positivo)
2245     self.label_limite_positivo_desfase3.setText(self.xml.
2246         lim_angulo_3_positivo)
2247     self.label_limite_positivo_desfase4.setText(self.xml.
2248         lim_angulo_4_positivo)
2249     self.label_limite_positivo_desfase5.setText(self.xml.
2250         lim_angulo_5_positivo)
2251
2252     # Configuraci n Zera
2253     for c in self.controles_configuracion_zera:
2254         self.fondo_azul_arteche(c)
2255         self.no_editable(c)
2256
2257     self.textEdit_Np.setText(PATRON_PRIMARIO)
2258     # tensi n primario del nuevo patron. Est multiplicado por raiz
2259     # de 3
2260     self.textEdit_Ns.setText(PATRON_SECUNDARIO)
2261     # tensi n secundaria del nuevo patron. Est multiplicado por raiz
2262     # de 3
2263     self.interpretacion_valor_frecuencia()
2264     self.textEdit_freq.setText(self.pz.frecuencia)
2265     self.textEdit_EVTp.setText(self.xml.num_t_prim.__str__())
2266     self.textEdit_EVTs.setText(self.xml.num_t_1_sec.__str__())
2267
2268     self.pz.set_evt_prim(self.textEdit_EVTp.toPlainText())
2269     self.pz.set_evt_sec(self.textEdit_EVTs.toPlainText())
2270     self.pz.set_n_prim(self.textEdit_Np.toPlainText())
2271     self.pz.set_n_sec(self.textEdit_Ns.toPlainText())
2272     self.pz.set_frecuencia(self.textEdit_freq.toPlainText())
2273
2274     # HIPOTRONICS
2275     self.mostrar_info_hipotronics()
2276
2277     # Limpiar datos ddpp
2278     self.limpiar_datos_ddpp()
2279
2280     def mostrar_info_ddpp(self):
2281         """Funci n que muestra la informaci n referente al ensayo de
2282             descargas parciales y configura el estilo de botones
2283             y casillas en la interfaz main.ui. Limpia lo referente al ensayo de
2284             precision.

```

```

2270
2271     Par metros:
2272         self — clase App
2273
2274     Atributos:
2275
2276         self.pushButton_prec — bot n PRECISION de la interfaz main.ui
2277             . Clase QPushButton de la librer a Qt5.
2278
2279         self.pushButton_DDPP — bot n DDPP de la interfaz main.ui.
2280             Clase QPushButton de la librer a Qt5.
2281
2282         self.controles_lecturas_zera — lista que hace referencia a los
2283             datos de las lecturas del puente Zera
2284
2285         self.controles_lecturas_precision — lista que hace referencia
2286             a los datos de las medidas de precision
2287
2288         self.controles_valores_error_modulo — lista que hace
2289             referencia a los errores en m dulo para ensayo de
2290             precisi n
2291
2292         self.controles_valores_error_angulo — lista que hace
2293             referencia a los errores en ngulo para ensayo de
2294             precisi n
2295
2296         self.textEdit_L1 — textEdit de la interfaz main.ui que hace
2297             referencia a la casilla del primer punto de
2298             tensi n a medir en %. Clase QTextEdit de la librer a Qt5.
2299
2300         self.textEdit_L2 — textEdit de la interfaz main.ui que hace
2301             referencia a la casilla del segundo punto de
2302             tensi n a medir en %. Clase QTextEdit de la librer a Qt5.
2303
2304         self.textEdit_L3 — textEdit de la interfaz main.ui que hace
2305             referencia a la casilla del tercer punto de
2306             tensi n a medir en %. Clase QTextEdit de la librer a Qt5.
2307
2308         self.textEdit_L4 — textEdit de la interfaz main.ui que hace
2309             referencia a la casilla del cuarto punto de
2310             tensi n a medir en %. Clase QTextEdit de la librer a Qt5.
2311
2312         self.textEdit_L5 — textEdit de la interfaz main.ui que hace
2313             referencia a la casilla del quinto punto de
2314             tensi n a medir en %. Clase QTextEdit de la librer a Qt5.
2315
2316         self.label_Error_Modulo_2 — label de la interfaz main.ui que
2317             hace referencia al valor del error en m dulo
2318             medido en el primer punto del ensayo de precisi n en %. Clase
2319             QLabel de la librer a Qt5.
2320
2321         self.label_Error_Modulo_3 — label de la interfaz main.ui que
2322             hace referencia al valor del error en m dulo
2323             medido en el segundo punto del ensayo de precisi n en %. Clase
2324             QLabel de la librer a Qt5.
2325
2326         self.label_Error_Modulo_4 — label de la interfaz main.ui que
2327             hace referencia al valor del error en m dulo

```



2312 *medido en el tercer punto del ensayo de precisión en %. Clase*  
2313 *QLabel de la librería Qt5.*

2314 *self.label\_Error\_Modulo\_5 — label de la interfaz main.ui que*  
2315 *hace referencia al valor del error en módulo*  
2316 *medido en el cuarto punto del ensayo de precisión en %. Clase*  
2317 *QLabel de la librería Qt5.*

2318 *self.label\_Error\_Modulo\_6 — label de la interfaz main.ui que*  
2319 *hace referencia al valor del error en módulo*  
2320 *medido en el quinto punto del ensayo de precisión en %. Clase*  
2321 *QLabel de la librería Qt5.*

2322 *self.label\_desfase\_2 — label de la interfaz main.ui que hace*  
2323 *referencia al valor del desfase medido en*  
2324 *el primer punto del ensayo de precisión en %. Clase QLabel de*  
2325 *la librería Qt5.*

2326 *self.label\_desfase\_3 — label de la interfaz main.ui que hace*  
2327 *referencia al valor del desfase medido en*  
2328 *el segundo punto del ensayo de precisión en %. Clase QLabel de*  
2329 *la librería Qt5.*

2330 *self.label\_desfase\_4 — label de la interfaz main.ui que hace*  
2331 *referencia al valor del desfase medido en*  
2332 *el tercer punto del ensayo de precisión en %. Clase QLabel de*  
2333 *la librería Qt5.*

2334 *self.label\_desfase\_5 — label de la interfaz main.ui que hace*  
2335 *referencia al valor del desfase medido en*  
2336 *el cuarto punto del ensayo de precisión en %. Clase QLabel de*  
2337 *la librería Qt5.*

2338 *self.label\_desfase\_6 — label de la interfaz main.ui que hace*  
2339 *referencia al valor del desfase medido en*  
2340 *el quinto punto del ensayo de precisión en %. Clase QLabel de*  
2341 *la librería Qt5.*

2342 *self.label\_tanda\_prec — label de la interfaz main.ui que hace*  
2343 *referencia a la tanda de ensayos del ensayo*  
2344 *de precisión. Clase QLabel de la librería Qt5.*

2345 *self.label\_tanda\_ddpp — label de la interfaz main.ui que hace*  
2346 *referencia a la tanda de ensayos del ensayo*  
2347 *de descargas parciales. Clase QLabel de la librería Qt5.*

2348 *self.controles\_limite\_negativo\_angulo — lista que hace*  
2349 *referencia a los límites negativos del error en*  
*ángulo para ensayo de precisión*

2350 *self.controles\_limite\_negativo\_modulo — lista que hace*  
2351 *referencia a los límites negativos del error en*  
2352 *módulo para ensayo de precisión*

2353 *self.controles\_limite\_positivo\_angulo — lista que hace*  
2354 *referencia a los límites positivos del error en*  
2355 *ángulo para ensayo de precisión*



2350 *self.controles\_limite\_positivo\_modulo* — lista que hace  
2351 referencia a los limites positivos del error en  
2352 modulo para ensayo de precision.  
2353 *self.controles\_lecturas\_ddpp* — lista que hace referencia a los  
2354 datos de las medidas de descargas parciales  
2355 *self.textEdit\_ddpp1* — *textEdit* de la interfaz *main.ui* que hace  
2356 referencia a la casilla del valor del ensayo  
de descargas parciales en el primer punto. Clase *QTextEdit* de  
la libreria *Qt5*.  
2357 *self.textEdit\_ddpp2* — *textEdit* de la interfaz *main.ui* que hace  
2358 referencia a la casilla del valor del ensayo  
de descargas parciales en el segundo punto. Clase *QTextEdit* de  
la libreria *Qt5*.  
2359 *self.textEdit\_ddpp3* — *textEdit* de la interfaz *main.ui* que hace  
2360 referencia a la casilla del valor del ensayo  
2361 de descargas parciales en el tercer punto. Clase *QTextEdit* de  
2362 la libreria *Qt5*.  
2363 *self.textEdit\_ddpp4* — *textEdit* de la interfaz *main.ui* que hace  
2364 referencia a la casilla del valor del ensayo  
2365 de descargas parciales en el cuarto punto. Clase *QTextEdit* de  
la libreria *Qt5*.  
2366 *self.textEdit\_ddpp5* — *textEdit* de la interfaz *main.ui* que hace  
2367 referencia a la casilla del valor del ensayo  
2368 de descargas parciales en el quinto punto. Clase *QTextEdit* de  
la libreria *Qt5*.  
2369 *self.controles\_limites\_ddpp* — lista que hace referencia a los  
2370 limites del error en descargas parciales  
2371 *self.label\_limite\_ddpp1* — *label* de la interfaz *main.ui* que  
2372 hace referencia al limite del valor del ensayo  
2373 de descargas parciales en el primer punto. Clase *QLabel* de la  
libreria *Qt5*.  
2374 *self.label\_limite\_ddpp2* — *label* de la interfaz *main.ui* que  
2375 hace referencia al limite del valor del ensayo  
2376 de descargas parciales en el segundo punto. Clase *QLabel* de la  
libreria *Qt5*.  
2377 *self.label\_limite\_ddpp3* — *label* de la interfaz *main.ui* que  
2378 hace referencia al limite del valor del ensayo  
2379 de descargas parciales en el tercer punto. Clase *QLabel* de la  
libreria *Qt5*.  
2380 *self.label\_limite\_ddpp4* — *label* de la interfaz *main.ui* que  
2381 hace referencia al limite del valor del ensayo  
2382 de descargas parciales en el cuarto punto. Clase *QLabel* de la  
libreria *Qt5*.  
2383 *self.label\_limite\_ddpp5* — *label* de la interfaz *main.ui* que  
2384 hace referencia al limite del valor del ensayo

```

2385         de descargas parciales en el quinto punto. Clase QLabel de la
2386             librería Qt5.
2387
2388         self.xml — clase ComunicacionWebService
2389
2390         self.pz — clase PuenteZera
2391
2392     """
2393     # BOTONES
2394     self.fondo_azul_arteche(self.pushButton_DDPP)
2395     self.fondo_blanco(self.pushButton_prec)
2396
2397     # Lecturas precisión
2398     for c in self.controles_lecturas_zera:
2399         self.fondo_gris(c)
2400     for c in self.controles_lecturas_precision:
2401         self.fondo_gris(c)
2402     for c in self.controles_valores_error_modulo:
2403         self.fondo_gris(c)
2404     for c in self.controles_valores_error_angulo:
2405         self.fondo_gris(c)
2406
2407     # Limites precisión
2408     for c in self.controles_limite_negativo_modulo:
2409         self.fondo_gris(c)
2410     for c in self.controles_limite_positivo_modulo:
2411         self.fondo_gris(c)
2412     for c in self.controles_limite_negativo_angulo:
2413         self.fondo_gris(c)
2414     for c in self.controles_limite_positivo_angulo:
2415         self.fondo_gris(c)
2416
2417     # Configuración Zera
2418     for c in self.controles_configuracion_zera:
2419         self.fondo_gris(c)
2420         self.no_editable(c)
2421
2422     # Info tanda
2423     self.fondo_naranja(self.label_tanda_ddpp)
2424     self.label_tanda_ddpp.setText(self.xml.NUM_TANDA + " " + self.xml
2425         .DESCRIPCION_TANDA)
2426
2427     self.fondo_gris(self.label_tanda_prec)
2428     self.texto_a_blanco(self.label_tanda_prec)
2429
2430     # DDPP
2431     for c in self.controles_lecturas_ddpp:
2432         self.editable(c)
2433         self.fondo_azul_arteche(c)
2434
2435     self.textEdit_ddpp1.setText(self.xml.valor_ddpp_1)
2436     self.textEdit_ddpp2.setText(self.xml.valor_ddpp_2)
2437     self.textEdit_ddpp3.setText(self.xml.valor_ddpp_3)
2438     self.textEdit_ddpp4.setText(self.xml.valor_ddpp_4)
2439     self.textEdit_ddpp5.setText(self.xml.valor_ddpp_5)
2440
2441     for c in self.controles_limites_ddpp:

```

```

2441         self.fondo_verde_arteche(c)
2442
2443     self.label_limite_ddpp1.setText(self.xml.lim_1_ddpp)
2444     self.label_limite_ddpp2.setText(self.xml.lim_2_ddpp)
2445     self.label_limite_ddpp3.setText(self.xml.lim_3_ddpp)
2446     self.label_limite_ddpp4.setText(self.xml.lim_4_ddpp)
2447     self.label_limite_ddpp5.setText(self.xml.lim_5_ddpp)
2448
2449     # HIPOTRONICS
2450     self.mostrar_info_hipotronics()
2451
2452     # Limpiar y deshabilitar ensayo ddpp
2453     self.limpiar_datos_precision()
2454
2455     def mostrar_info_hipotronics(self):
2456         """Funci n que muestra la informaci n del Hipotronics en la
2457             interfaz main.ui
2458
2459             Par metros:
2460                 self — clase App
2461
2462             Atributos:
2463                 self.ensayo — tipo de ensayo seleccionado que puede ser
2464                     precisi n o descargas parciales
2465
2466                 self.textEdit_Rampa1 — textEdit de la interfaz main.ui que
2467                     hace referencia a la casilla del valor de la
2468                     inclinaci n de la rampa del Hipotronics del primer punto.
2469                     Clase QTextEdit de la librer a Qt5.
2470
2471                 self.textEdit_Rampa2 — textEdit de la interfaz main.ui que
2472                     hace referencia a la casilla del valor de la
2473                     inclinaci n de la rampa del Hipotronics del segundo punto.
2474                     Clase QTextEdit de la librer a Qt5.
2475
2476                 self.textEdit_Rampa3 — textEdit de la interfaz main.ui que
2477                     hace referencia a la casilla del valor de la
2478                     inclinaci n de la rampa del Hipotronics del tercer punto.
2479                     Clase QTextEdit de la librer a Qt5.
2480
2481                 self.textEdit_Rampa4 — textEdit de la interfaz main.ui que
2482                     hace referencia a la casilla del valor de la
2483                     inclinaci n de la rampa del Hipotronics del cuarto punto.
2484                     Clase QTextEdit de la librer a Qt5.
2485
2486                 self.textEdit_Rampa5 — textEdit de la interfaz main.ui que
2487                     hace referencia a la casilla del valor de la
2488                     inclinaci n de la rampa del Hipotronics del quinto punto.
2489                     Clase QTextEdit de la librer a Qt5.
2490
2491                 self.textEdit_Tension1 — textEdit de la interfaz main.ui que
2492                     hace referencia a la casilla del valor de la
2493                     tensi n de la rampa del Hipotronics en el primer punto. Clase
2494                     QTextEdit de la librer a Qt5.
2495
2496                 self.textEdit_Tension2 — textEdit de la interfaz main.ui que
2497                     hace referencia a la casilla del valor de la
2498                     tensi n de la rampa del Hipotronics en el segundo punto. Clase
2499                     QTextEdit de la librer a Qt5.

```

```

2483     tensi n de la rampa del Hipotronics en el segundo punto. Clase
2484         QTextEdit de la librer a Qt5.
2485
2486     self.textEdit_Tension3 — textEdit de la interfaz main.ui que
2487         hace referencia a la casilla del valor de la
2488         tensi n de la rampa del Hipotronics en el tercer punto. Clase
2489             QTextEdit de la librer a Qt5.
2490
2491     self.textEdit_Tension4 — textEdit de la interfaz main.ui que
2492         hace referencia a la casilla del valor de la
2493         tensi n de la rampa del Hipotronics en el cuarto punto. Clase
2494             QTextEdit de la librer a Qt5.
2495
2496     self.textEdit_Tension5 — textEdit de la interfaz main.ui que
2497         hace referencia a la casilla del valor de la
2498         tensi n de la rampa del Hipotronics en el quinto punto. Clase
2499             QTextEdit de la librer a Qt5.
2500
2501     self.textEdit_Tiempo1 — textEdit de la interfaz main.ui que
2502         hace referencia a la casilla del tiempo de
2503         espera de la rampa del Hipotronics en el primer punto. Clase
2504             QTextEdit de la librer a Qt5.
2505
2506     self.textEdit_Tiempo2 — textEdit de la interfaz main.ui que
2507         hace referencia a la casilla del tiempo de
2508         espera de la rampa del Hipotronics en el segundo punto. Clase
2509             QTextEdit de la librer a Qt5.
2510
2511     self.textEdit_Tiempo3 — textEdit de la interfaz main.ui que
2512         hace referencia a la casilla del tiempo de
2513         espera de la rampa del Hipotronics en el tercer punto. Clase
2514             QTextEdit de la librer a Qt5.
2515
2516     self.textEdit_Tiempo4 — textEdit de la interfaz main.ui que
2517         hace referencia a la casilla del tiempo de
2518         espera de la rampa del Hipotronics en el cuarto punto. Clase
2519             QTextEdit de la librer a Qt5.
2520
2521     self.textEdit_Tiempo5 — textEdit de la interfaz main.ui que
2522         hace referencia a la casilla del tiempo de
2523         espera de la rampa del Hipotronics en el quinto punto. Clase
2524             QTextEdit de la librer a Qt5.
2525
2526     self.controles_hipotronics — lista que hace referencia a los
2527         datos del Hipotronics
2528
2529
2530     """
2531     if self.ensayo == ENSAYO_PRECISION:
2532         self.textEdit_Tiempo1.setText("5")
2533         self.textEdit_Tiempo2.setText("5")
2534         self.textEdit_Tiempo3.setText("5")
2535         self.textEdit_Tiempo4.setText("5")
2536         self.textEdit_Tiempo5.setText("5")
2537
2538         self.definir_puntos_tension_hipotronics_precision()
2539
2540     elif self.ensayo == ENSAYO_DDPP:

```

```

2523     self.textEdit_Tiempo1.setText("60")
2524     self.textEdit_Tiempo2.setText("20")
2525     self.textEdit_Tiempo3.setText("20")
2526     self.textEdit_Tiempo4.setText("20")
2527     self.textEdit_Tiempo5.setText("20")
2528
2529     self.definir_puntos_tension_hipotronics_ddpp()
2530
2531     self.textEdit_Rampa1.setText(self.h.ramp1)
2532     self.textEdit_Rampa2.setText(self.h.ramp2)
2533     self.textEdit_Rampa3.setText(self.h.ramp3)
2534     self.textEdit_Rampa4.setText(self.h.ramp4)
2535     self.textEdit_Rampa5.setText(self.h.ramp5)
2536
2537     self.textEdit_Tension1.setText(self.h.targetVoltage1.__str__())
2538     self.textEdit_Tension2.setText(self.h.targetVoltage2.__str__())
2539     self.textEdit_Tension3.setText(self.h.targetVoltage3.__str__())
2540     self.textEdit_Tension4.setText(self.h.targetVoltage4.__str__())
2541     self.textEdit_Tension5.setText(self.h.targetVoltage5.__str__())
2542
2543     self.definir_datos_hipotronics()
2544     # Interfaz
2545     for c in self.controles_hipotronics:
2546         self.fondo_azul_arteche(c)
2547         self.no_editable(c)
2548
2549     # Limpiar datos
2550     def limpiar_datos_precision(self):
2551         """Funci n que limpia todas las casillas y botones referentes al
2552             ensayo de precisi n de la interfaz main.ui
2553
2554             Par metros:
2555                 self — clase App
2556
2557             Atributos:
2558                 self.label_tanda_prec — label de la interfaz main.ui que hace
2559                 referencia a la tanda de ensayos del ensayo
2560                 de precisi n. Clase QLabel de la librer a Qt5.
2561
2562                 self.pushButton_prec — bot n PRECISION de la interfaz main.ui
2563                 . Clase QPushButton de la librer a Qt5.
2564
2565                 self.controles_lecturas_zera — lista que hace referencia a los
2566                 datos de las lecturas del puente Zera
2567
2568                 self.controles_lecturas_precision — lista que hace referencia
2569                 a los datos de las medidas de precision
2570
2571                 self.controles_valores_error_modulo — lista que hace
2572                 referencia a los errores en m dulo para ensayo de
                 precisi n
2573
2574                 self.controles_valores_error_angulo — lista que hace
                 referencia a los errores en ngulo para ensayo de
                 precisi n
2575
2576                 self.controles_limite_negativo_angulo — lista que hace
                 referencia a los limites negativos del error en

```

```

2573         ngulo para ensayo de precisi n
2574
2575     self.controles_limite_negativo_modulo — lista que hace
2576         referencia a los limites negativos del error en
2577         m dulo para ensayo de precisi n
2578
2579     self.controles_limite_positivo_angulo — lista que hace
2580         referencia a los limites positivos del error en
2581         ngulo para ensayo de precisi n
2582
2583     self.controles_limite_positivo_modulo — lista que hace
2584         referencia a los limites positivos del error en
2585         m dulo para ensayo de precisi n
2586
2587     """
2588     # Informaci n tanda
2589     self.fondo_gris(self.label_tanda_prec)
2590     self.texto_a_blanco(self.label_tanda_prec)
2591
2592     # Botones
2593     self.fondo_blanco(self.pushButton_prec)
2594
2595     for c in self.controles_lecturas_zera:
2596         self.fondo_gris(c)
2597         self.texto_a_blanco(c)
2598
2599     # controles_lecturas_precision
2600     for c in self.controles_lecturas_precision:
2601         self.fondo_gris(c)
2602         self.texto_a_blanco(c)
2603
2604     # controles_valores_error_modulo
2605     for c in self.controles_valores_error_modulo:
2606         self.fondo_gris(c)
2607         self.texto_a_blanco(c)
2608
2609     # controles_valores_error_angulo
2610     for c in self.controles_valores_error_angulo:
2611         self.fondo_gris(c)
2612         self.texto_a_blanco(c)
2613
2614     # controles_limite_negativo_modulo
2615     for c in self.controles_limite_negativo_modulo:
2616         self.fondo_gris(c)
2617         self.texto_a_blanco(c)
2618
2619     # controles_limite_positivo_modulo
2620     for c in self.controles_limite_positivo_modulo:
2621         self.fondo_gris(c)
2622         self.texto_a_blanco(c)
2623
2624     # controles_limite_negativo_angulo
2625     for c in self.controles_limite_negativo_angulo:
2626         self.fondo_gris(c)
2627         self.texto_a_blanco(c)

```

```

2627
2628     # controles_limite_positivo_angulo
2629     for c in self.controles_limite_positivo_angulo:
2630         self.fondo_gris(c)
2631         self.texto_a_blanco(c)
2632
2633     # controles_configuracion_zera
2634     for c in self.controles_configuracion_zera:
2635         self.fondo_gris(c)
2636         self.texto_a_blanco(c)
2637         self.no_editable(c)
2638
2639     def limpiar_datos_ddpp(self):
2640         """Funci n que limpia todas las casillas y botones referentes al
2641             ensayo de descargas parciales de la interfaz main.ui
2642
2643             Par metros:
2644                 self — clase App
2645
2646             Atributos:
2647                 self.label_tanda_ddpp — label de la interfaz main.ui que hace
2648                 referencia a la tanda de ensayos del ensayo
2649                 de descargas parciales. Clase QLabel de la librer a Qt5.
2650
2651                 self.pushButton_DDPP — bot n DDPP de la interfaz main.ui.
2652                 Clase QPushButton de la librer a Qt5.
2653
2654                 self.controles_lecturas_ddpp — lista que hace referencia a los
2655                 datos de las medidas de descargas parciales
2656
2657                 self.controles_limites_ddpp — lista que hace referencia a los
2658                 limites del error en descargas parciales
2659
2660                 """
2661         # Quitar color y texto a la info de la tanda
2662         self.fondo_gris(self.label_tanda_ddpp)
2663         self.texto_a_blanco(self.label_tanda_ddpp)
2664
2665         # botones
2666         self.fondo_blanco(self.pushButton_DDPP)
2667
2668         for c in self.controles_lecturas_ddpp:
2669             self.fondo_gris(c)
2670             self.texto_a_blanco(c)
2671             self.no_editable(c)
2672
2673         # controles_limites_ddpp
2674         for c in self.controles_limites_ddpp:
2675             self.fondo_gris(c)
2676             self.texto_a_blanco(c)
2677
2678     def limpiar_info_general(self):
2679         """Funci n que limpia lo referente a la informaci n general en la
2680             interfaz main.ui.
2681
2682             Par metros:
2683                 self — clase App

```

```

2679     Atributos:
2680         self.controles_info_general — lista que hace referencia a la
                información general de la OF
2681
2682         self.xml — clase ComunicacionWebService
2683     """
2684     # controles_info_genera
2685     for c in self.controles_info_general:
2686         self.fondo_gris(c)
2687         self.texto_a_blanco(c)
2688
2689     # inicializar atributos del xml
2690     self.xml.nombre_cliente = ""
2691     self.xml.orden_fab = ""
2692     self.xml.OV_IN_OUT = ""
2693     self.xml.modelo = ""
2694     self.xml.temp_max = ""
2695     self.xml.temp_min = ""
2696     self.xml.CLASE = ""
2697     self.xml.factor_tension = ""
2698     self.xml.norma = ""
2699     self.xml.carga_capac = ""
2700     self.xml.carga_resist = ""
2701
2702     # Pantallas secundarias
2703     @staticmethod
2704     def mostrar_mensaje_error(titulo, mensaje):
2705         """Función que enseña un mensaje de error .
2706
2707         Parametros:
2708             self — clase App
2709             titulo — str texto a mostrar como título de la ventana
2710             mensaje — str texto del mensaje a mostrar
2711
2712         Atributos:
2713             msg_box — clase QMessageBox
2714
2715         """
2716         msg_box = QtWidgets.QMessageBox()
2717         msg_box.setIcon(QtWidgets.QMessageBox.Warning)
2718         msg_box.setWindowTitle(titulo)
2719         msg_box.setText(mensaje)
2720
2721         msg_box.exec()
2722
2723     @staticmethod
2724     def pantalla_log_in():
2725         """Función que llama a la pantalla de log in y devuelve el usuario
                y la contraseña a que ha escrito el usuario.
2726
2727         Atributo:
2728             ventana_log_in — clase VentanaLogIn
2729
2730         Devuelve:
2731             input_dialog — lista que contiene el usuario y la contraseña a
                introducimos por el usuario
2732
2733         """

```



```

2734     input_dialog = venatan_log_in.main()
2735     return input_dialog
2736
2737     @staticmethod
2738     def mostrar_mensaje_ensayado():
2739         """Funci n que crea una ventana que indica que la orden ya tiene
2740             valores guardados. Pregunta si se desea crear
2741             una nueva tanda y da dos opciones al usuario: Yes/No
2742
2743             Atributo:
2744                 msg_box — clase QMessageBox
2745
2746             Devuelve:
2747                 16384 — al clicar Yes
2748                 65536 — al clicar No
2749
2750             """
2751         # Crear ventana
2752         msg_box = QtWidgets.QMessageBox()
2753         msg_box.setIcon(QtWidgets.QMessageBox.Question)
2754         msg_box.setWindowTitle("Valores ya guardados")
2755         msg_box.setText("La orden ya tiene valores guardados en el ensayo
2756             solicitado. Desea crear una nueva tanda?")
2757         msg_box.addButton(QtWidgets.QMessageBox.Yes)
2758         msg_box.addButton(QtWidgets.QMessageBox.No)
2759
2760         # Ejecutar ventana y devolver el valor del bot n clicado (Yes:
2761             16384 No:65536)
2762         return msg_box.exec()
2763
2764     @staticmethod
2765     def mostrar_ventana_nueva_tanda():
2766         """Funci n que crea una ventana para elegir la raz n por la cual
2767             se quiere realizar una nueva tanda de ensayo.
2768
2769             Atributo:
2770                 msg_box — clase QMessageBox
2771
2772             Devuelve:
2773                 ventana_tanda — la opci n elegida
2774
2775             """
2776         # Crear ventana
2777         ventana_tanda = QtWidgets.QInputDialog()
2778         ventana_tanda.setWindowTitle("Nueva tanda")
2779
2780         # Crear combo con las opciones de la tanda
2781         ventana_tanda.setComboBoxItems(OPCIONES_NUEVA_TANDA)
2782         ventana_tanda.setLabelText("Raz n para la tanda:")
2783
2784         # Ejecutar ventana
2785         ventana_tanda.exec()
2786         return ventana_tanda
2787
2788     def show_dialog_error_usuario_contrasena(self):
2789         """Funci n que visualiza un mensaje de error de introducci n de
2790             usuario y contrase a.

```

```

2787     Parametros:
2788         self — clase App
2789
2790     Atributos:
2791         logger — Logger
2792
2793     """
2794     self.mostrar_mensaje_error("ERROR", "Alg n dato ha sido
2795         introducido incorrectamente")
2796     self.logger.exception("Error usuario y contrase a")
2797
2798 def show_dialog_error_lectura_datos(self):
2799     """Funci n que visualiza un mensaje de error de lectura de datos
2800         del web service
2801
2802     Parametros:
2803         self — clase App
2804
2805     Atributos:
2806         logger — Logger
2807
2808     """
2809     self.mostrar_mensaje_error("ERROR", "No se han podido leer los
2810         datos correctamente")
2811     self.logger.exception("Error lectura de datos")
2812
2813 # ESTILOS
2814 @staticmethod
2815 def texto_a_blanco(*args):
2816     """Funci n que pone en blanco el texto del argumento que se le da.
2817
2818     Par metros
2819         *args — tupla compuesta por uno o varios argumentos de
2820             cualquier tipo
2821
2822     """
2823     for control in args:
2824         control.setText("")
2825
2826 @staticmethod
2827 def no_editable(*args):
2828     """Funci n que pone en modo no editable el argumento que se le da.
2829
2830     Par metros
2831         *args — tupla compuesta por uno o varios argumentos de
2832             cualquier tipo
2833
2834     """
2835     for control in args:
2836         control.setReadOnly(True)
2837
2838 @staticmethod
2839 def editable(*args):
2840     """Funci n que pone en modo editable el argumento que se le da.
2841
2842     Par metros
2843         *args — tupla compuesta por uno o varios argumentos de
2844             cualquier tipo

```

```

2839
2840     """
2841     for control in args:
2842         control.setReadOnly(False)
2843
2844     @staticmethod
2845     def fondo_azul_arteche(*args):
2846         """Funci n que le pone el fondo azul al argumento que se le da.
2847
2848         Par metros
2849         *args — tupla compuesta por uno o varios argumentos de
2850             cualquier tipo
2851
2852         """
2853         for control in args:
2854             control.setStyleSheet(u"background-color: rgb(0, 154, 197);\n"
2855                                   "font: 75 14pt\''Ms Gotham Book\'';\n")
2856
2857     @staticmethod
2858     def fondo_naranja(*args):
2859         """Funci n que le pone el fondo naranja al argumento que se le da.
2860
2861         Par metros
2862         *args — tupla compuesta por uno o varios argumentos de
2863             cualquier tipo
2864
2865         """
2866         for control in args:
2867             control.setStyleSheet(u"background-color: rgb(235, 127, 19);\n"
2868                                   "font: 75 8pt\''Ms Gotham Book\'';\n")
2869
2870     @staticmethod
2871     def fondo_verde_arteche(*args):
2872         """Funci n que le pone el fondo verde al argumento que se le da.
2873
2874         Par metros
2875         *args — tupla compuesta por uno o varios argumentos de
2876             cualquier tipo
2877
2878         """
2879         for control in args:
2880             control.setStyleSheet(u"background-color: rgb(197,216,130);\n"
2881                                   "font: 75 8pt\''Ms Gotham Book\'';\n")
2882
2883     @staticmethod
2884     def fondo_gris(*args):
2885         """Funci n que le pone el fondo gris al argumento que se le da.
2886
2887         Par metros
2888         *args — tupla compuesta por uno o varios argumentos de
2889             cualquier tipo
2890
2891         """
2892         for control in args:
2893             control.setStyleSheet(u"background-color: rgb(154, 161, 150);\n"
2894                                   "font: 75 12pt\''Ms Gotham Book\'';\n")
2895
2896     @staticmethod

```

```
2889 def fondo_blanco(*args):
2890     """Funci n que le pone el fondo blanco al argumento que se le da.
2891
2892     Par metros
2893     *args — tupla compuesta por uno o varios argumentos de
                cualquier tipo
2894
2895     """
2896     for control in args:
2897         control.setStyleSheet(u"background-color: white;\n"font: 75 14
                pt\ "Ms Gotham Book\";\n")
2898
2899 @staticmethod
2900 def fondo_rojo(*args):
2901     """Funci n que le pone el fondo rojo al argumento que se le da.
2902
2903     Par metros
2904     *args — tupla compuesta por uno o varios argumentos de
                cualquier tipo
2905
2906     """
2907     for control in args:
2908         control.setStyleSheet(u"background-color: red;")
2909
2910
2911 # -----
2912 # Programa principal
2913 # -----
2914 if __name__ == "__main__":
2915     app = QtWidgets.QApplication(sys.argv)
2916     window = App()
2917     window.showNormal()
2918     sys.exit(app.exec_())
```

## F. VentanaLogIn Klasea

```

1 from tkinter import Tk, BOTH, X, LEFT
2 from tkinter.ttk import Frame, Label, Entry, Button
3
4
5 class VentanaLogIn(Frame):
6     """Clase de un pop up dialog simple para interfaz.
7
8     Ventana para la seguridad del programa. Pide dos datos: Usuario y
9     Contrase a.
10
11     Devuelve una tupla de dos elementos str (usuario,password).
12
13     """
14     def __init__(self):
15         """Funcion por defecto de inicializacion de objeto dentro de la
16         clase. Se les asignan valores de inicializaci n a los atributos
17         del objeto.
18
19         Crea mediante la funci n de inicializaci n super de la clase Ttk
20         Frame un Frame estandar y lo personaliza mediante la funci n
21         init_ui().
22
23         Parametros:
24             self — clase VentanaLogIn
25
26         Atributos:
27             self.usuario — el usuario que se va a utilizar para el uso del
28             web service que nos proporcionara datos de ensayo
29
30             self.password — password ligada al usuario amteriormente
31             mencionado.
32
33         """
34         super().__init__()
35         self.usuario = ""
36         self.contrasena = ""
37         self.init_ui()
38         # Defini c i n de las variables de entrada
39
40     def init_ui(self):
41         """Funci n para construir y personalizar la ventana de seguridad.
42
43         Paramentros:
44             self — clase VentanaLogIn
45
46         Atributos:
47             frame1 — Frame secundario para el atributo self.usuario
48
49             lbl1 — Label para el atributo self.usuario
50
51             self.entry1 — variable de entrada para el atributo sel.
52             usuario
53
54             frame2 — Frame secundario para el atributo self.password
55
56             lbl2 — Label para el atributo self.password

```

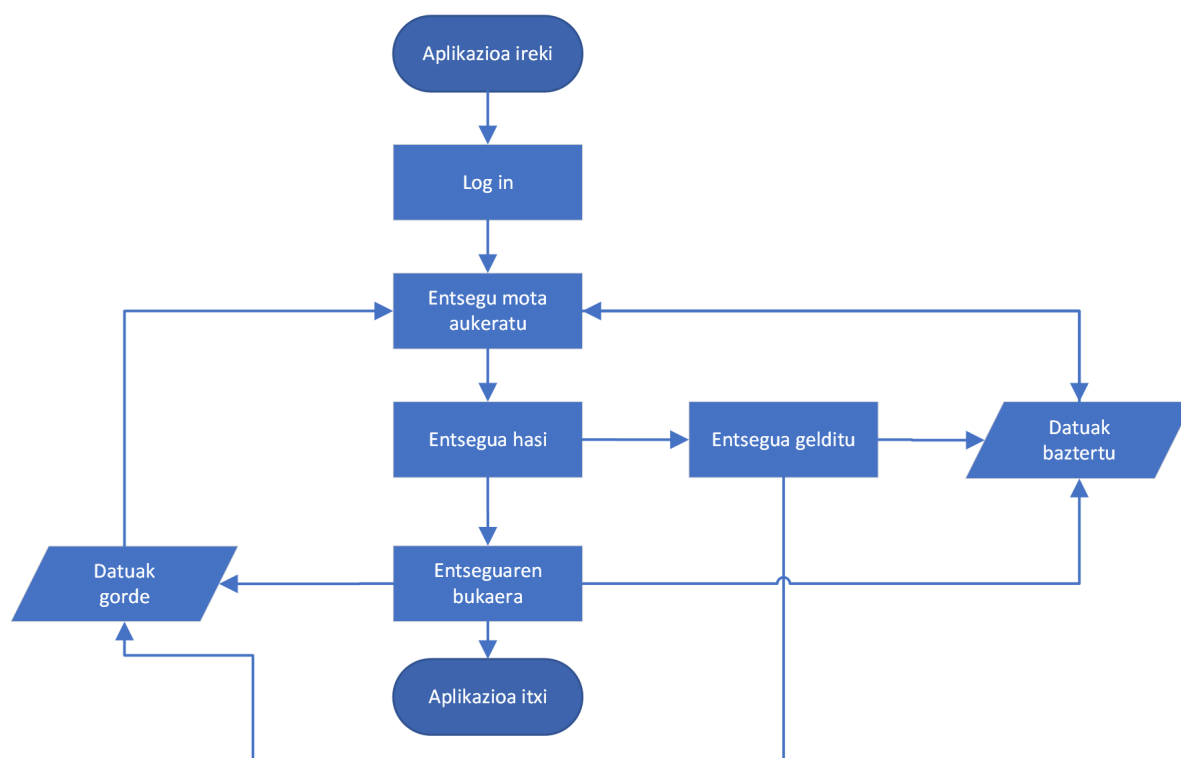
```

49
50         self.entry2 — variable de entrada para el atributo sel.
           password
51
52         frame3 — Frame secundario para el atributo btn
53
54         btn — bot n que llama a la funci n self.on_submit
55             """
56
57     self.master.title("SEGURIDAD")
58     self.pack(fill=BOTH, expand=True)
59
60     frame1 = Frame(self)
61     frame1.pack(fill=X)
62
63     lbl1 = Label(frame1, text="USUARIO", width=13)
64     lbl1.pack(side=LEFT, padx=5, pady=10)
65
66     self.entry1 = Entry(frame1, textvariable=self.usuario)
67     self.entry1.pack(fill=X, padx=5, expand=True)
68
69     frame2 = Frame(self)
70     frame2.pack(fill=X)
71
72     lbl2 = Label(frame2, text="CONTRASEA", width=13)
73     lbl2.pack(side=LEFT, padx=5, pady=10)
74
75     self.entry2 = Entry(frame2)
76     self.entry2.pack(fill=X, padx=5, expand=True)
77     self.entry2.config(show="*")
78
79     frame3 = Frame(self)
80     frame3.pack(fill=X)
81
82     btn = Button(frame3, text="Submit", command=self.on_submit)
83     btn.pack(padx=5, pady=10)
84
85     def on_submit(self):
86         """Funci n que se activa al darle al bot n Submit.
87
88         Funci n que recoge lo que el usuario ha escrito en los frames
           referentes a Usuario y Contrase a y los guarda en los atributos
           self.usuario y self.password.
89
90         Al finalizar llama al m todo self.quit() para salir de la ventana.
91
92         Parametros:
93             self — clase VentanaLogIn
94
95         """
96         self.usuario = self.entry1.get()
97         self.contrasena = self.entry2.get()
98         self.quit()
99
100
101     def main():
102         """Funci n principal para la creaci n de la clase y que devuelve el
           usuarion y la password.

```

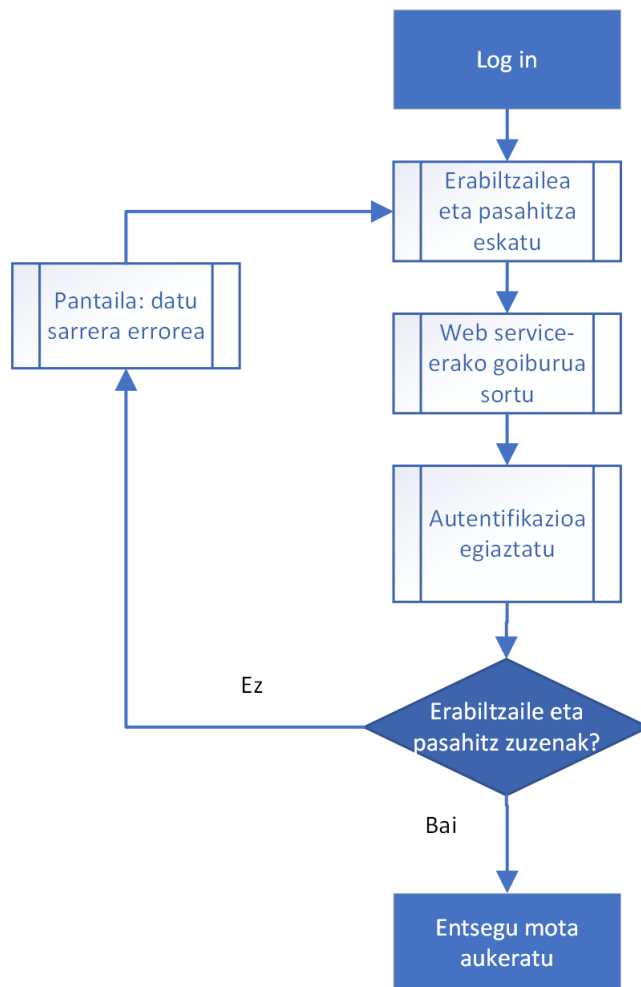
```
103
104 Atributos:
105     root — Clase Tk
106
107     app — clase VentanaLogIn
108
109 Parametros de salida:
110     user_input — tupla que recoge el valor de4 usuario en la primera
111     posici n y el de la password en el segundo.
112
113 Excepciones:
114     Si cualquier excepci n ocurre al intentar cerrar la ventana se
115     recoge y se pasa.
116
117 """
118 root = Tk()
119 root.geometry("250x150+300+300")
120 app = VentanaLogIn()
121 root.mainloop()
122 user_input = (app.usuario , app.contrasena)
123 try:
124     root.destroy()
125 except:
126     pass
127 return user_input
```

## G. Fluxu-diagramak

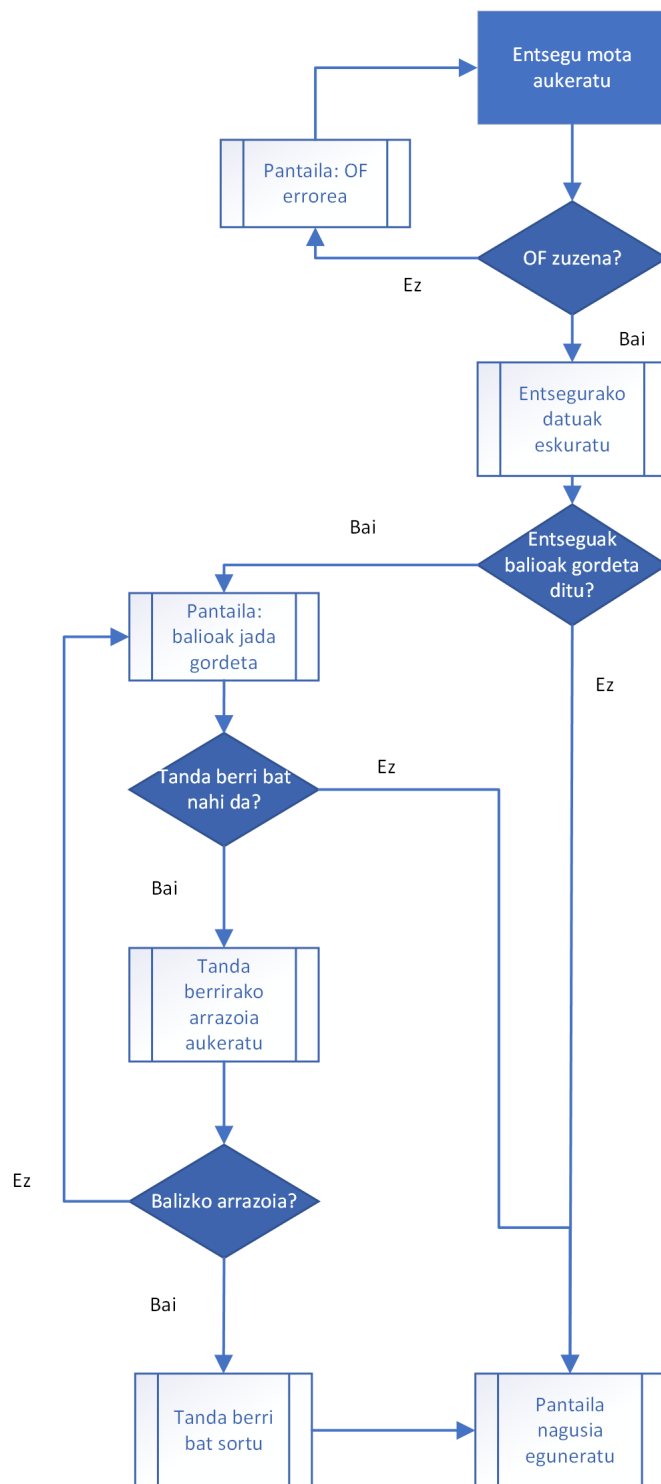


**Fluxu-diagrama 1:** programa nagusiaren fluxu-diagrama

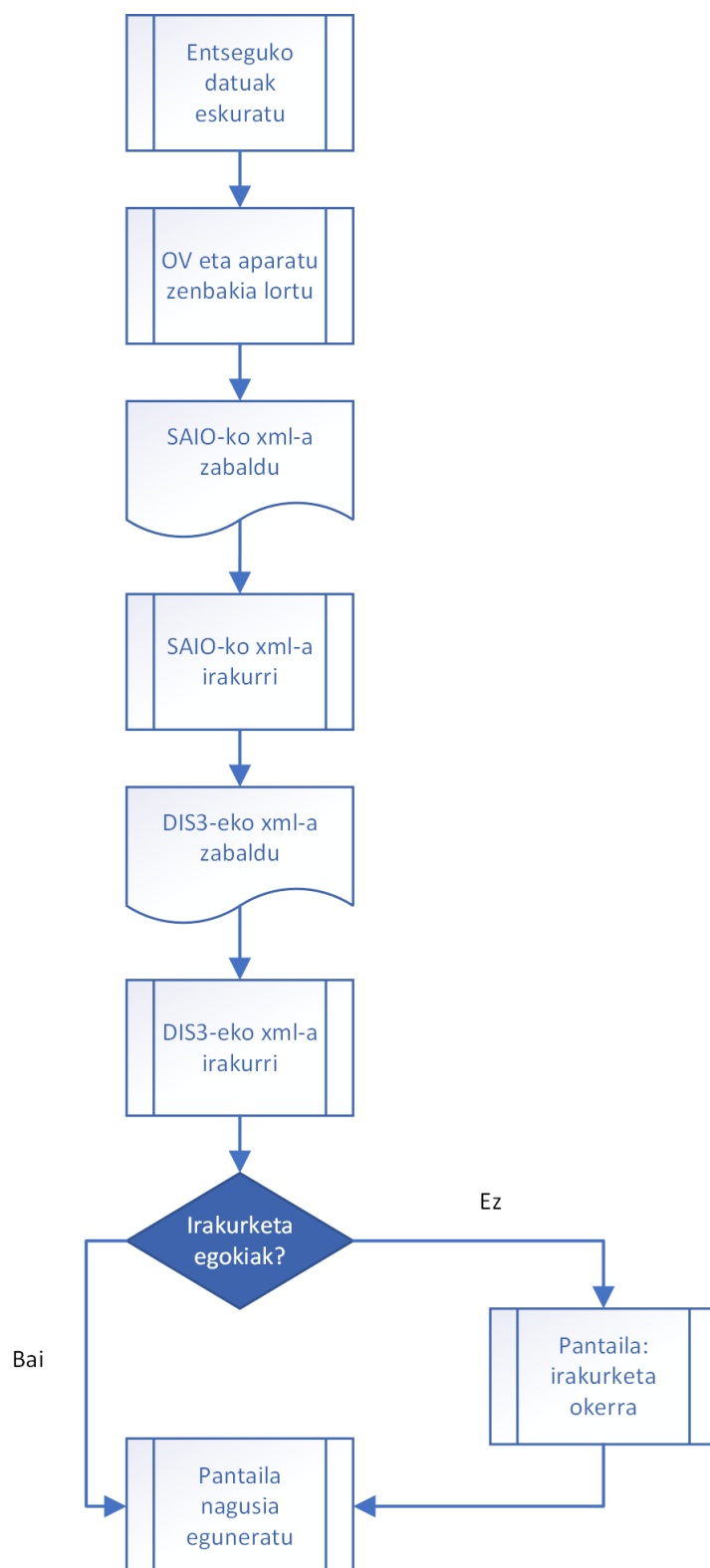




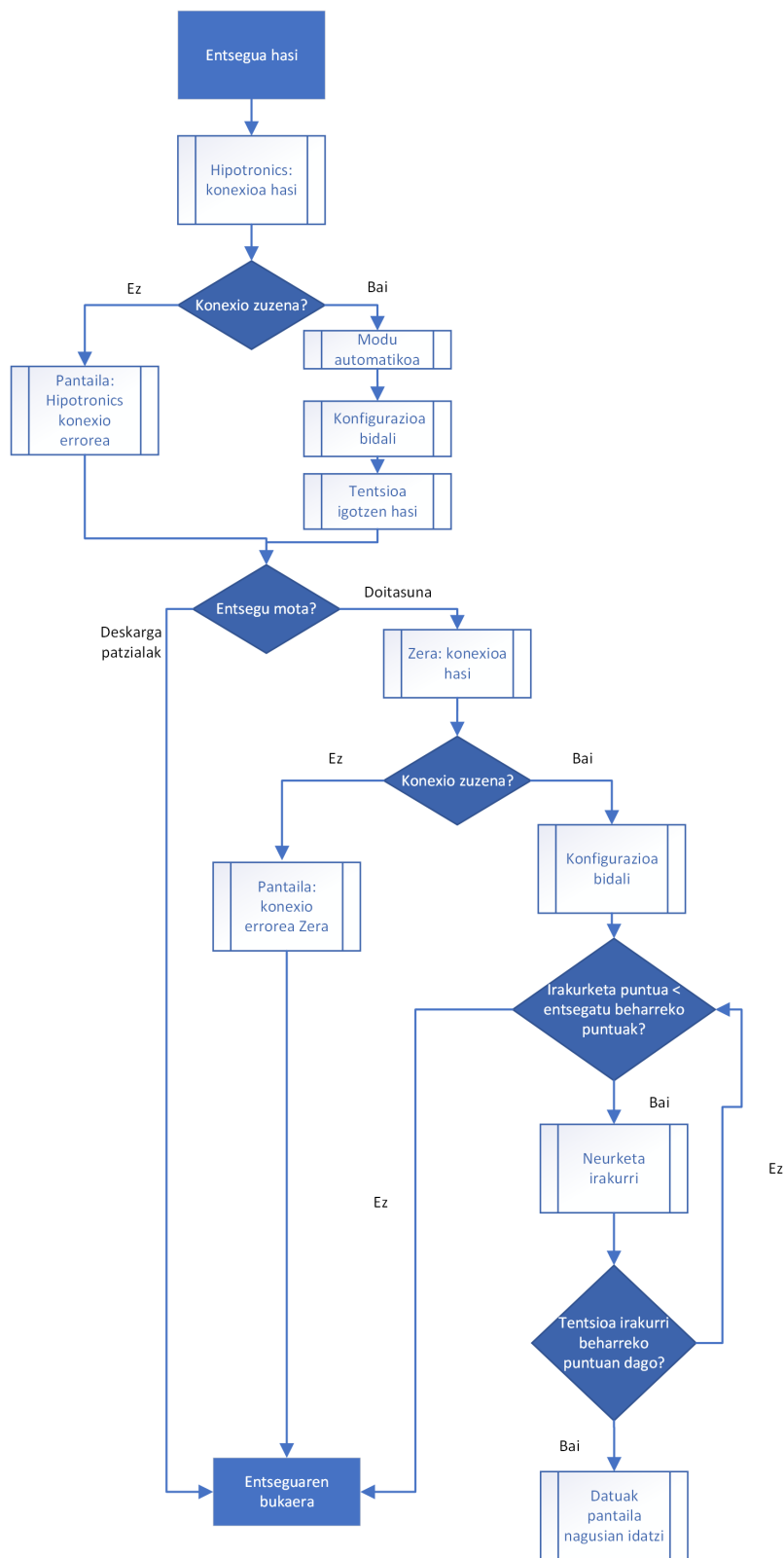
**Fluxu-diagrama 2:** *Log in azpiprozesuaren fluxu diagrama*



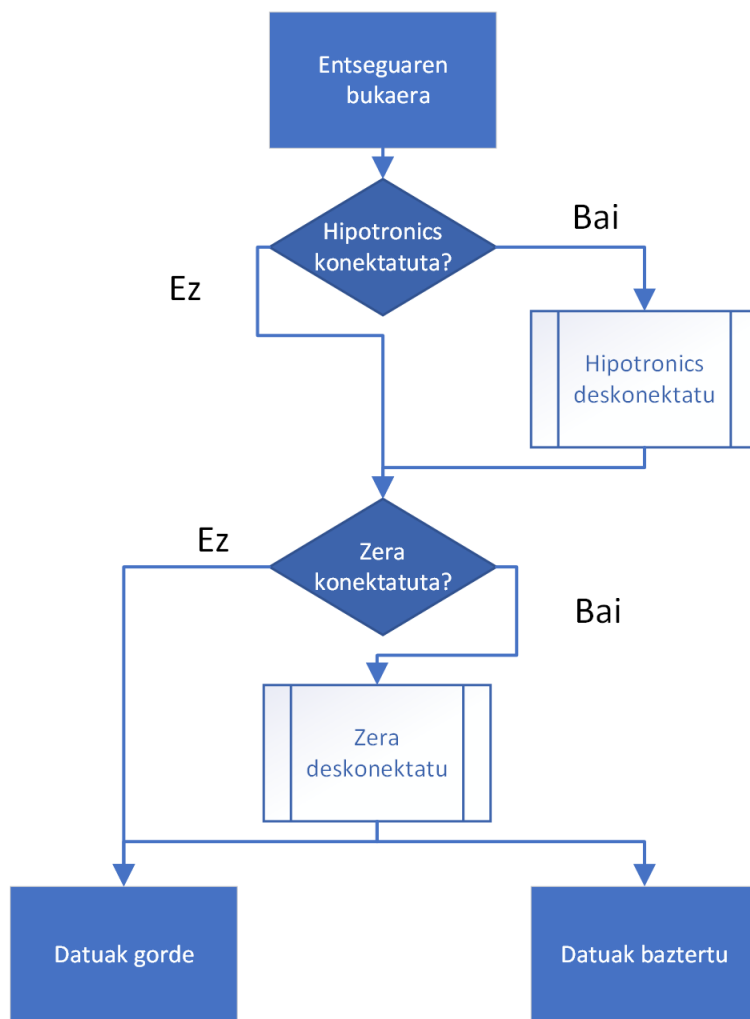
**Fluxu-diagrama 3:** entsegu aukeraketa egitean martxan jartzen den prozesua. Barnean dauka entsegurako datuak eskuratzeko jarraitzen duen azpi-prozesua Fluxu-diagrama 4-en ageri dena.



**Fluxu-diagrama 4:** *datu baseetatik datuak irakurtzeko azpi-prozesua*



Fluxu-diagrama 5: entsegua hasi prozesuaren fluxu-diagrama



**Fluxu-diagrama 6:** entsegua bukatzeko prozesuaren fluxu-diagrama