agronomy

MDPI

Article

Development of a Low-Cost Open-Source Platform for Smart Irrigation Systems

Francisco Puig *, Juan Antonio Rodríguez Díaz and María Auxiliadora Soriano

Department of Agronomy, University of Cordoba, Campus Rabanales, 14014 Cordoba, Spain
* Correspondence: g32pupef@uco.es

Abstract: Nowadays, smart irrigation is becoming an essential goal in agriculture, where water and energy are increasingly limited resources. Its importance will grow in the coming years in the agricultural sector where the optimal use of resources and environmental sustainability are becoming more important every day. However, implementing smart irrigation is not an easy task for most farmers since it is based on knowledge of the different processes and factors that determine the crop water requirements. Thanks to technological developments, it is possible to design new tools such as sensors or platforms that can be connected to soil-water-plant-atmosphere models to assist in the optimization and automation of irrigation. In this work, a low-cost, open-source IoT system for smart irrigation has been developed that can be easily integrated with other platforms and supports a large number of sensors. The platform uses the FIWARE framework together with customized components and can be deployed using edge computing and/or cloud computing systems. To improve decision-making, the platform integrates an irrigation model that calculates soil water balance and wet bulb dimensions to determine the best irrigation strategy for drip irrigation systems. In addition, an energy efficient open-source datalogger has been designed. The datalogger supports a wide range of communications and is compatible with analog sensors, SDI-12 and RS-485 protocols. The IoT system has been deployed on an olive farm and has been in operation for one irrigation season. Based on the results obtained, advantages of using these technologies over traditional methods are discussed.

Keywords: IoT platform; precision agriculture; irrigation model; FIWARE; soil water balance; dual crop coefficient

check for updates

Citation: Puig, F.; Rodríguez Díaz, J.A.; Soriano, M.A. Development of a Low-Cost Open-Source Platform for Smart Irrigation Systems. *Agronomy* **2022**, *12*, 2909. https://doi.org/10.3390/agronomy12122909

Academic Editor: Juan Manuel Díaz-Cabrera

Received: 28 September 2022
Accepted: 18 November 2022
Published: 22 November 2022

1. Introduction

In Spain, as in the rest of southern Europe, water is a scarce, fragile, and unevenly distributed resource [1]. Climate models developed by the European Environment Agency (EEA) [2] show that, in the last 50 years, in the Iberian Peninsula, there has been a decrease in precipitation of up to 90 mm per decade, and predict that in the next 70 years, this decrease will be even greater, reaching a reduction of up to 40% in some areas of southern Spain. As a result of decreased rainfall and overexploitation of water resources, the balance between available water and water demand has already reached critical levels.

In the context of this worrying future scenario, solutions need to be found. Improving irrigation efficiency and optimizing agricultural management practices are two of the necessary actions, and technology is an indispensable tool to carry them out.

Precision agriculture (PA) has emerged as an approach to optimize farm management, which involves the use of a series of sensors and actuators that allow gathering context information from the surrounding environment [3].

Using different types of sensors and algorithms, the irrigation requirements of crops can be calculated automatically so that the amount of water they need can be estimated and applied at the right time.

This development has led to an increase in the number of companies offering platforms for the visualization of sensor data, which, while useful, still requires specialized farmers who know how to interpret the data and apply the most appropriate management strategies. In addition, these platforms, being mostly commercial, are characterized as closed systems, which do not allow connection with third-party sensors and often have high costs that prevent the farmer from installing several measurement points, which is a problem in the analysis of spatial variability within the farm.

Several IoT (Internet of Things) platforms for agriculture can be found in the literature. Previous researchers [4–6] created an open IoT platform developed with FIWARE [7] for agriculture, but none of them use the linked data (LD) version that allows a greater standardization of the API (application programming interface), which is essential for the platform to have open data. Platform [8] uses the LD version of FIWARE and proposes a data model to improve the management and real-time monitoring of crops in irrigation communities. Additionally, an IoT platform based on microservices, such as FIWARE, but with its own architecture, so it does not have a standardized API, has been proposed [9]. However, none of the above works integrates soil-water-plant-atmosphere models. [10] propose a mobile application that incorporates an agricultural model [11] for fertilization and irrigation scheduling, but this model does not use sensor data and is based on the single crop coefficient (K_c) approach instead of the dual K_c for separate estimation of crop transpiration (T_c) and soil evaporation (E_s) (i.e., $K_c = K_{cb} + K_e$), introduced in FAO56 [12] which improves the accuracy of potential crop evapotranspiration (ET_c) estimation.

To solve these problems, a low-cost IoT platform for precision irrigation has been developed in this work. Using FIWARE, more specifically the LD version, the objective is to create an open IoT application with an architecture that allows the connection of any type of sensor and agronomic model and with a standardized API that allows platforms of the same type, which also use FIWARE, to connect to each other without the need for additional configurations. The platform not only works as a dashboard, displaying the data captured by the sensors, but also automatically calculates the crop water needs using the dual Kc approach and allows the implementation of deficit irrigation strategies. To connect different sensors to the platform, a low-cost, energy-efficient data acquisition device has been developed.

2. Materials and Methods

2.1. Study Area

This project has been carried out on a farm devoted to olive groves located in Córdoba, southern Spain.

The climate in this area is characterized by mild winters and hot summers, with mean daily temperatures around 10 °C and above 26 °C, respectively (Table 1). The monthly mean values of grass reference evapotranspiration (ET_0; FAO-Penman-Monteith method [12]) varies between 1.1 and 7.6 mm day^{-1}.

The average annual rainfall and ET_0 for the period from 2001 to 2021 was 590 mm, with high interannual variability, and 1409 mm, respectively (Figure 1). The year 2021, when the project started, was a dry year, with an annual rainfall of 436 mm.

The soil is alkaline (pH = 8.7), with a high presence of carbonates (45.3%) and active limestone (18.2%). The texture varied between loam and clay loam, with a medium water retention capacity. The soil had no salinity problems, with an average electrical conductivity (EC; 1:5) value of 0.14 dS/m. The average cation exchange capacity (CEC) was 24 cmol(+)/kg, with the exchange complex dominated by the presence of Ca and a high K and low Mg content. The total organic matter content of soil is 1.43%.

This olive farm consists of two plots of olive trees. The first of the plots is made up of olive trees of the Arbequina variety, planted in 2017 with a planting frame of 6 × 2 m. The second plot is made up of olive trees of the Arbosana variety, planted in 2019 with a planting frame of 5 × 1.75 m. Both plots are irrigated by means of a pump that drives the water from the Guadajoz River to the different irrigation sectors. Table 2 shows the characteristics of the irrigation system.

Table 1. Average climatic data in Córdoba (Andalusia; southern Spain) during the period from 2001 to 2021, obtained from the Córdoba agroclimatic station belonging to the Andalusian Agroclimatic Information Network (RIA).

Month	**Precipitation (mm month^{-1})**	**Mean T (°C)**	**Mean Relative Humidity (%)**	**Wind Speed (m s^{-1})**	**Solar Radiation (MJ m^{-2} d^{-1})**	***ET_0* (mm day $^{-1}$)**
January	56.8	8.4	81.9	1.38	8.7	1.17
February	63.3	10.1	76.7	1.55	11.8	1.79
March	79.0	13.0	71.9	1.65	16.1	2.79
April	62.5	15.9	68.3	1.62	20.3	3.80
May	36.3	19.9	58.1	1.62	24.8	5.20
June	7.5	24.8	48.1	1.90	27.9	6.78
July	2.2	27.9	40.9	1.99	28.3	7.61
August	10.7	27.9	42.2	1.85	25.1	6.82
September	32.0	23.7	54.6	1.63	19.3	4.67
October	78.2	18.6	67.1	1.45	13.7	2.82
November	76.5	12.2	77.3	1.39	9.7	1.57
December	84.9	9.2	81.5	1.41	7.8	1.11
Year	590	17.7	64.1	1.62	17.8	3.86

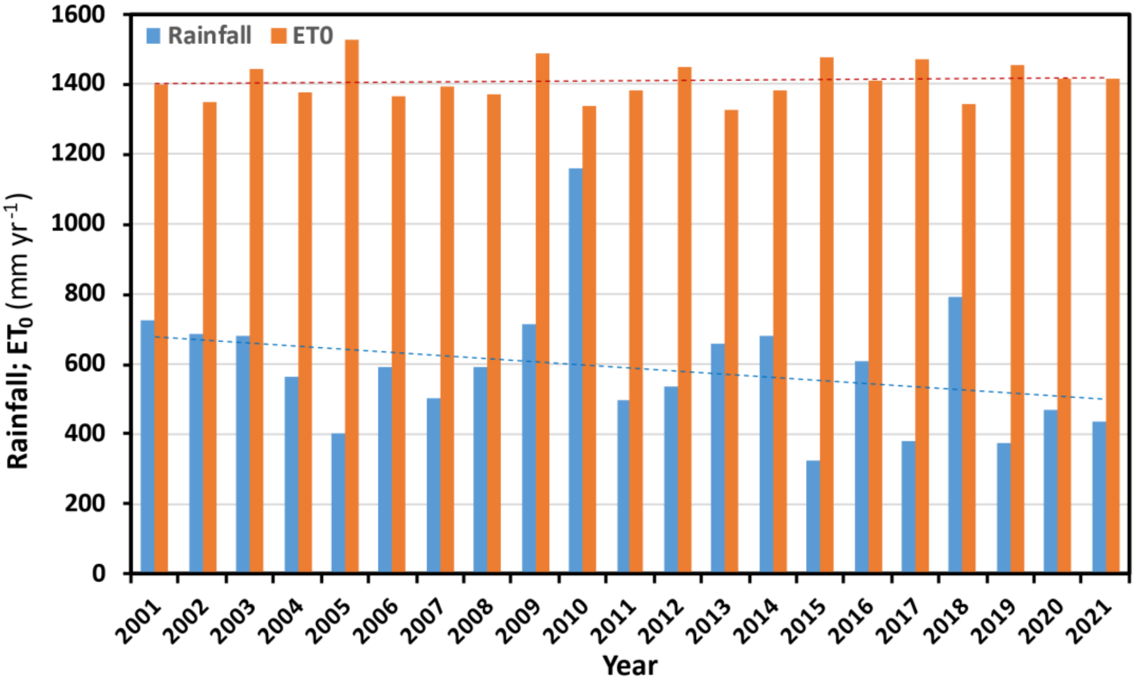

Figure 1. Annual precipitation and *ET_0* data recorded at the Cordoba agroclimatic station belonging to the RIA, from 2001 to 2021.

Table 2. Characteristics of the irrigation sectors. Sector one with the Arbequina variety and sector two with the Arbosana variety.

	Area (ha)	**Dripper Flow Rate (L/h)**	**Drip Spacing (m)**	**Irrigation Efficiency (%)**
Sector one	22.7	2.2	0.7	90
Sector two	19.3	2.2	0.7	90

2.2. Iot Platform

2.2.1. General Architecture

The architecture of the developed IoT platform (Figure 2) is divided into three distinct and independent layers that are interconnected through a set of APIs. This architecture is made up of a set of microservices that perform a specific task. The central component of the platform is the Orion Context Broker, which is responsible for connecting the different microservices using the same API. This type of architecture increases the scalability and security of the system as it allows the creation of new services that perform specific functionalities without affecting those already in place.

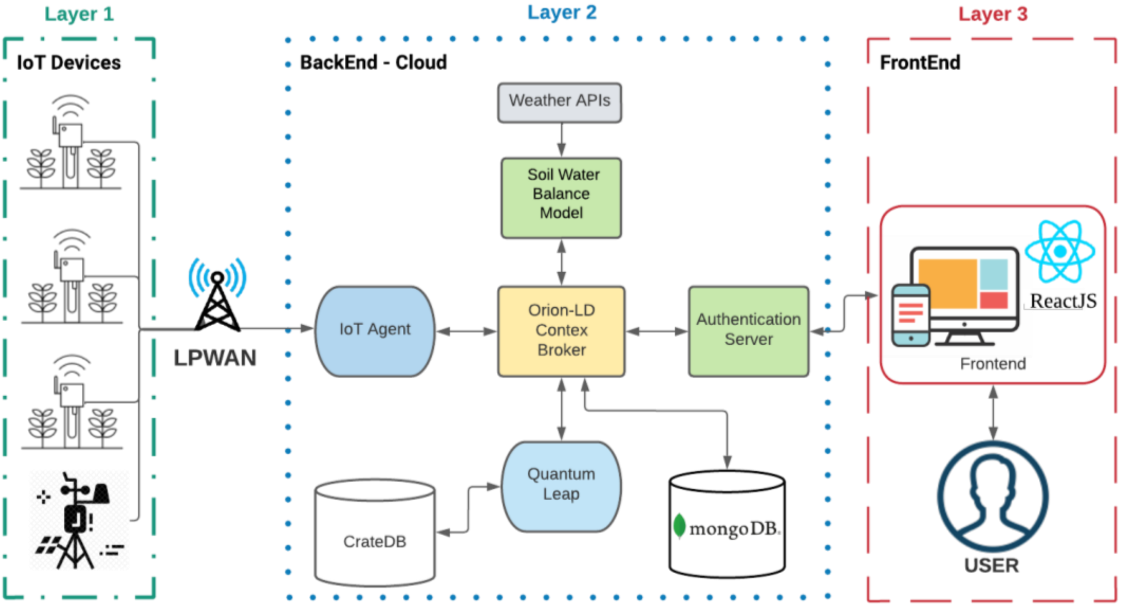

Figure 2. IoT Platform Architecture.

2.2.2. Layer 1 (IoT Devices)

All IoT applications need to have one or more sensors to collect data from the environment. Sensors are essential components of smart objects. One of the most important aspects of the Internet of Things is context awareness, which is not possible without sensor technology [13]. The first layer is made up of smart and energy-efficient devices that have been developed with the aim of being compatible with different connections and protocols. These devices have been developed to accomplish the following characteristics:

- Wireless and with different communication protocols: The devices are usually installed in dispersed locations without internet connectivity and have therefore been developed to support a wide range of communications such as Low Power Wide Area Networks (LPWAN), including LoRaWAN, SigFox, NB-IoT, Global System for Mobile (GSM), Wi-Fi, and Bluetooth. Depending on the coverage and the number of sensors (e.g., LoRa network needs to deploy an infrastructure that is not cost-effective for a limited number of sensors), one communication system or another will be chosen.
- Energy-efficient: Sensors in agriculture are usually installed in remote locations that are difficult to access, so they must be energy self-sufficient.
- Supports different output protocols: To measure the different agronomic factors affecting irrigation, the devices have to support the main protocols used by the sensors. These protocols are SDI-12 (Serial Digital Interface at 1200 baud), RS-485 (Communication standard published by the Telecommunications Industry Association and Electronic Industries Alliance; TIA/EIA), I2C (Inter-Integrated Circuit), and SPI (Serial Peripheral Interface). The devices also support analog sensors.

The devices are programmed with the open-source Arduino software, which can be extended with C and C++ programming languages.

Hardware Architecture

The devices are made up of a set of components that together form an embedded system. The main element of the devices is the microcontroller, which monitors the control of power consumption, obtains data from the sensors, and communicates with layer two. For this purpose, the Arduino MKR family of boards has been used, which are open source and support a wide range of communication protocols. This board is based on Atmel's 32-bit SAMD21 microcontroller.

To integrate this board with the sensors, a printed circuit board (PCB) has been developed. A schematic is shown in Figure 3. The PCB can be divided into the following parts, each of which performs a specific functionality.

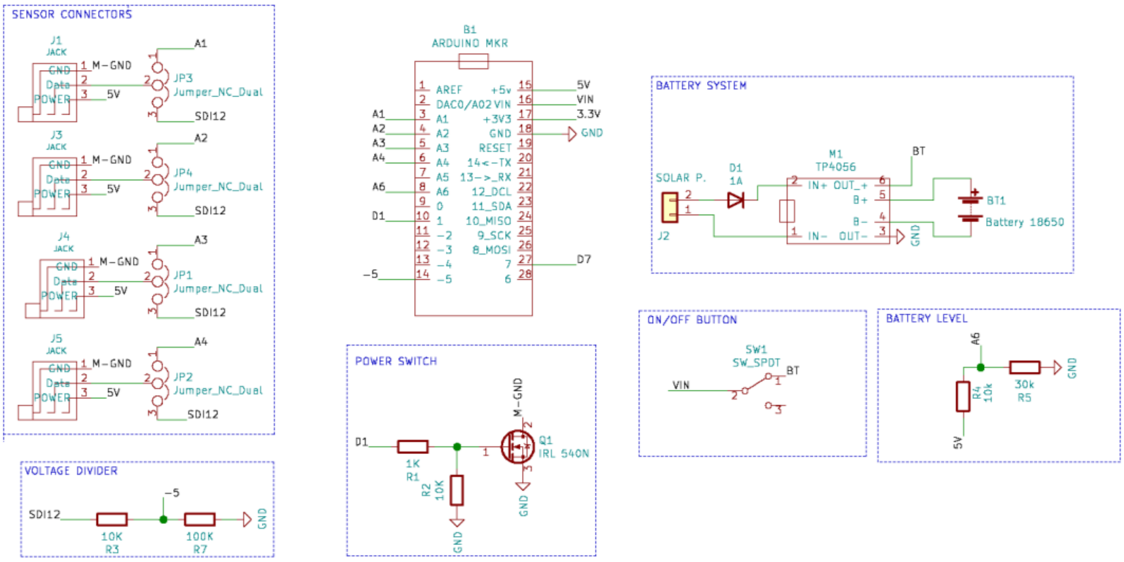

Figure 3. Device Schematic.

- Battery system: consists of a 2600 mAh, 3.7 V, rechargeable 18,650 lithium-ion battery, which is charged by a 250 mA solar panel. To control the battery charge, the Lion TP4056 linear charger module has been used. This module controls the overcharging and discharging of the battery.
- Power Switch: It consists of a Logic Level Mosfet, which is used as a switch, cutting off the power supply to the sensors during the sleep period and supplying it at the moment of taking measurements. This reduces power consumption considerably.
- Voltage Divider: adjust the battery voltage to the MKR circuit operating voltage of 3.3 V.
- On/off button: button that turns the device on and off.
- Sensor Connectors: It has four jack connectors for easy integration with sensors. These have a jumper that enables switching from digital to analog.

2.2.3. Layer 2 (IoT Backend)

The second layer is the backend. It is based on a microservices architecture and uses the FIWARE framework alongside custom services. FIWARE brings a curated framework of open source platform components that can be assembled together and with other third-party platform components to build smart solutions faster, easier, and cheaper [7]. It is supported by the European Union (EU) and has been developed to create open source, standardized IoT platforms that are suitable for all sectors, such as smart cities, industry, and agriculture, and that enable better communication between systems. The EU's commitment to FIWARE is due to the need to create a common standard in the IOT world, as until then each organization followed different paths that prevented communication between sensors and platforms. With FIWARE, this problem is solved, and cities, industries, or agricultural farms follow the same standard that allows them to share information in a simple way.

FIWARE is composed of a series of software blocks, called generic enabler (GE), which perform specific functions. Each of these GEs provides open APIs, which facilitate cross-platform communication and make it easier to develop IoT applications. This is one of the main advantages offered by this framework since it allows integration between different FIWARE-based platforms. This is very important both at the farm management and research levels because it allows the integration of methods developed by research groups and facilitates their use by farmers and the private sector, improving decision making.

In FIWARE, the state of an object in the real world is defined as context information and is represented by an entity. Each entity has attributes that are divided into properties and relationships, where the first defines the value of the entity and the latter defines the relationship that this entity has with other entities. For example, in a "Device" entity, the "SoilMoisture" attribute is a property, while the "Parcel" attribute is a relationship linking this sensor to the crop plot where it is located. The values of this attribute represent the current status of the device, which is the context information, and only the latest state of the device is stored in the context broker. Therefore, to access historical data, it is necessary to connect to external services that store the information in databases.

The components of the backend are:

- Orion Context Broker with Linked Data Extensions (Orion-LD): It represents the main component of the platform and enables the management of the entire lifecycle of context information, including updates, queries, registrations, and subscriptions [14]. This component is based on the NGSI-LD API (Next Generation Service Interfaces—Linked Data) [15] that supports the LD concept.
- QuantumLeap: a GE that is responsible for persistently storing temporal sensor data in a time-series database.
- IoT Agent: a GE that allows the connection of the sensors with the context broker. It manages the different devices and is responsible for transforming the sensor-specific protocol into the NGSI-LD context information protocol.
- Authentication Server: a custom microservice that is responsible for authentication and authorization of users to the platform. It is developed with Node.js, a JavaScript runtime environment, and uses JSON Web Tokens to ensure security.
- Soil Water Balance Model: A customized microservice that calculates the soil water balance of crop plots on a daily basis. The model has been developed with the Python programming language, and the microservice for accessing the model with Node.js. This model is described in more detail in Section 2.3.
- MongoDB: a No-SQL database that stores context data and user information.
- CrateDB: another No-SQL database where the sensor data is stored in a persistent way.

To connect all the services or components that are in the backend, the NGSI-LD API provides two mechanisms called subscriptions and registrations. The first one informs the external services that the context data has changed, and the second one connects the services to the context broker, so that the context broker changes the context data. For example, there is a subscription that connects Orion-LD to the QuantumLeap service. This subscription aims to redirect to QuantumLeap the sensor data coming to the broker (via the IoT-Agent). An example of registration is when an external service, in this case the IoT-Agent, wants to connect to the broker to change the context data; this IoT-Agent has an associated registration, which is responsible for changing the status of the sensors in the broker when data from the sensor is received again.

Linked Data

Linked data concepts allow for one step further in the communication between platforms as it uses the JSON-LD (JavaScript Object Notation for Linked Data) data format, which makes it possible to read and write structured data supported by open vocabularies [16] that can be read by computers. This makes it easier to find and exchange information with open databases, mobile apps, and IoT platforms. This is especially useful in the IoT, where information exchange between platforms is currently very limited.

The Orion-LD broker version, currently in beta, allows users to manage and request context information in an organized way using the NGSI-LD standards. For example, as will be seen in the following section, the application developed in this article has an entity called "Parcel", which represents a crop plot within the farm. This same entity within the FIWARE standards, which are publicly available, is called "AgriParcel". To relate these two concepts, FIWARE makes use of JSON-LD files, so that when our platform asks FIWARE to return the entitiy "Parcel", FIWARE knows that it is referring to the "AgriParcel" entity. As these standards are unique and public, different platforms can access the same FIWARE API using their own entity names, which may be different between applications.

Figure 4 shows how two apps can access different FIWARE servers using their own entities.

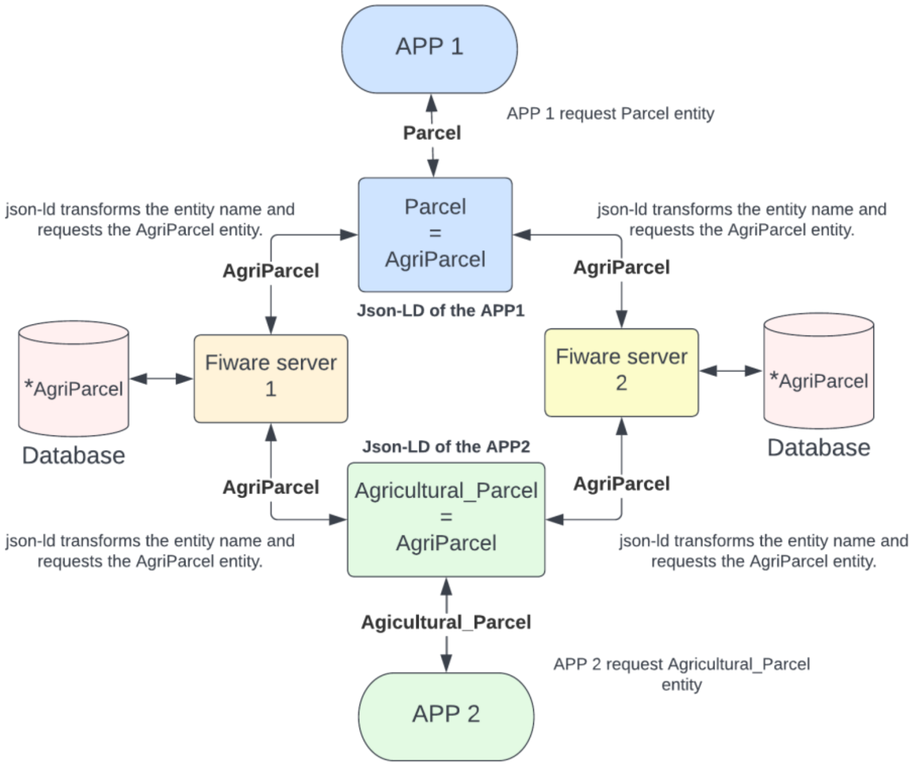
APP 1
APP 1 request Parcel entity
Parcel
json-ld transforms the entity name and requests the AgriParcel entity.
Parcel = AgriParcel
json-ld transforms the entity name and requests the AgriParcel entity.
AgriParcel
AgriParcel
Json-LD of the APP1
*AgriParcel
Fiware server 1
Fiware server 2
*AgriParcel
Database
Database
Json-LD of the APP2
AgriParcel
Agricultural_Parcel = AgriParcel
AgriParcel
json-ld transforms the entity name and requests the AgriParcel entity.
json-ld transforms the entity name and requests the AgriParcel entity.
Agicultural_Parcel
APP 2 request Agricultural_Parcel entity
APP 2

Figure 4. Simplified diagram of how JSON-LD works in FIWARE (with Orion-LD) using the "AgriParcel" entity.

2.2.4. Layer 3 (Frontend)

The last layer is the frontend, which can be defined as the web design and development technologies that allow the user to interact with the backend (Layer 2). This layer consists of a user interface that acts as an intermediary between the user and the platform, displaying information and facilitating user interaction with the platform. To connect to the Rest API of the backend, the frontend uses the HTTP protocol.

For the development of this layer, the ReactJS library has been used, which is based on HTML5, JavaScript, and CSS technologies. This library allows the creation of single-page applications (SPA), which significantly increases user experience and application performance. The web application consists of a single HTML document, and as the user interacts, the content changes dynamically.

The frontend, as well as serving as an interface to facilitate the user's connection with the server, is responsible for structuring the application. It is where the entities will be structured so that all users will follow the same pattern (Figure 5). The main entity in the application is the farm, where general aspects such as location, name, owner, etc. are collected. Within the farms are the entities ("Parcel") corresponding to the crop plots. These entities contain information about the agronomic factors such as soil type, crop characteristics, irrigation system, irrigation strategy, and user management.

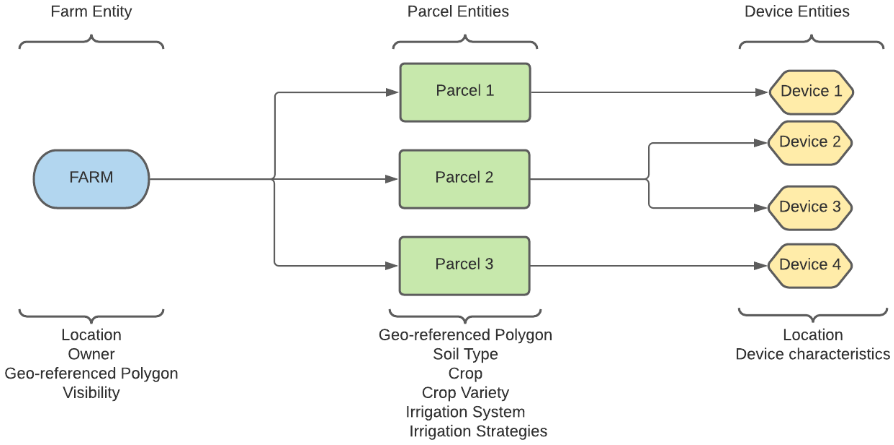

Figure 5. Structure of the entities.

Finally, within each "Parcel", there are the entities corresponding to the devices. These entities represent the dataloggers installed in the farm and show the information and historical data of each of the devices. This structure makes it possible to simulate any farm in which there are parcels with different agronomic characteristics, on each of which one or more sensors are installed.

Each user is assigned a role with specific permissions. Users with the role of administrator are the only ones authorized to add or delete users and administer the databases. The other users can only manage those entities that they have created themselves.

Users can make the farms they have created public or give permission to different users, so that any of them can see the data but cannot edit anything. In addition, each farm has a private API-Key that can be shared to access the NGSI-LD API.

2.3. Irrigation Model

The irrigation model is responsible for calculating the amount of water that the crop needs and when the best time is to apply it. Although it could be used for any irrigation method, in this particular case it is focused on drip irrigation as it allows the calculation of the wet bulb dimensions.

Figure 6 shows the different elements that make up the irrigation model developed in this article. This diagram can be divided into two parts: the first one related to the soil water balance model, which is calculated using the FAO-56 methodology [17], and the second one related to the wet bulb, where the total available soil water (TAW) and the wet bulb dimensions are calculated.

The application runs two types of models. The first one calculates the soil water balance from previous years. The user can choose whether to make a water balance by simulating a specific year or by making the balance using the average of the climatic values of the last 20 years. As the user can change the irrigation strategies, this first balance allows them to obtain the results of each of the strategies that they define and find the one that best suits their needs. The second type uses weather forecasts to estimate the crop water requirements in the next seven days and apply the most efficient irrigation strategy possible. All the weather data can be obtained using an on-farm weather station or an external meteorological API, such as AEMET and elTiempo.es, two meteorological services operating in Spain.

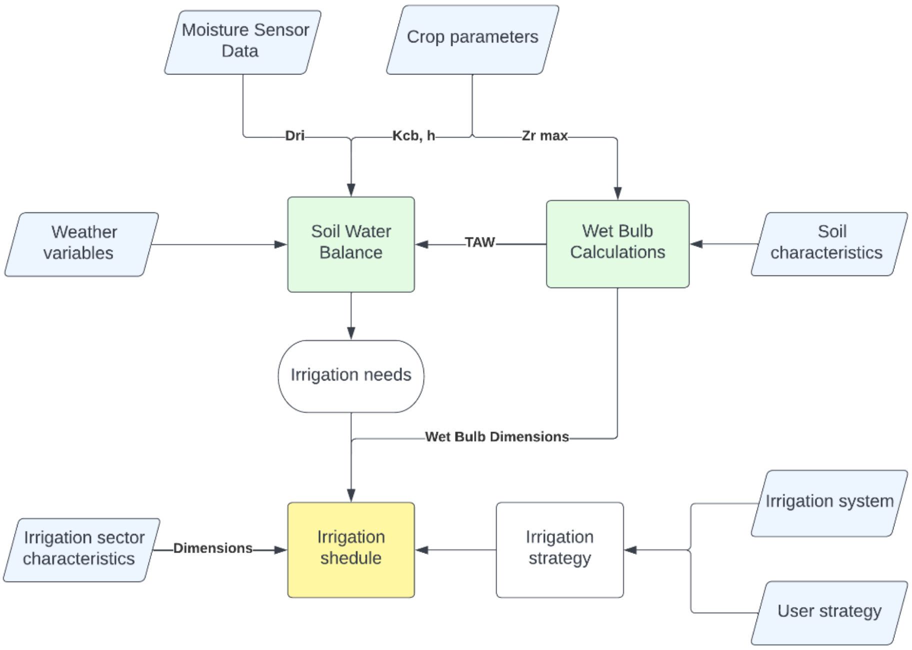

Figure 6. General outline of the proposed irrigation model. AEMET is the Spanish State Meteorological Agency, and elTiempo.es is a Spanish website where weather forecasts are shown.

2.3.1. Soil Water Balance

The soil water balance (SWB) model uses daily climatic data, crop characteristics, and data recorded by sensors to calculate crop water consumption (ET_a) and water losses in percolation and runoff processes.

The daily SWB, expressed as water depletion at the end of day i, is calculated using the following equation:

$$D_{r,i} = D_{r,i-1} - (P - RO)_i - I_i - CR_i + ET_{a,i} + DP_i \quad (1)$$

where $D_{r,i}$ is the soil water depletion in the root zone at the end of day i [mm]; $D_{r,i-1}$ is the depletion in the root zone at the end of the previous day $i - 1$ [mm]; P_i is the precipitation on day i [mm]; RO_i is the runoff from the soil surface on day i [mm]; I_i is the irrigation applied on day i [mm]; CR_i is the capillary rise from the groundwater table on day i [mm]; $ET_{a,\ i}$ is actual crop evapotranspiration on day i [mm], and DP_i is the water flowing out from the root zone -by deep percolation on day i [mm].

ET_a was calculated following the FAO56 method [17] as the product of the reference evapotranspiration (ET_0), calculated using the FAO Penman-Monteith equation, and the actual crop coefficient $(K_{c\ act})$, $(ET_a = ET_0 \times K_{c\ act})$, with $K_{c\ act} = K_s \times K_c$, where K_s is a crop water stress coefficient, and K_c is calculated using the FAO56 dual K_c method ($K_c = K_{cb} + K_e$) [12]. Where K_{cb} is the basal crop coefficient and K_e is and evaporation coefficient, which represent the ratios of crop transpiration (T_c) and soil evaporation (E_s), respectively (T_c/ET_0, and E_s/ET_0) [18]. Thus, $K_{c\ act}$ is calculated as follows:

$$K_{c\ act} = K_s K_{cb} + K_e \quad (2)$$

To adopt the dual K_c approach, the methodology used in the SIMDualKc software has been used [19]. This methodology has been validated in high-density olive groves [20–22] and in other woody row crops, such as peach trees [23] and vineyards [24].

For the calculation of K_{cb}, the differences in canopy density and crop height have to be considered as these parameters influence crop transpiration. K_{cb} can be calculated using the following equation [25]:

$$K_{cb} = K_{cb\ min} + K_d\left(K_{cb\ full} - K_{cb\ min}\right) \tag{3}$$

where $K_{cb\ full}$ represents the estimated K_{cb} at maximum crop growth, for conditions having nearly full ground cover, $K_{cb\ min}$ is the minimum K_{cb} for bare soil ($K_{cb\ min} \sim 0.15$) [19] and K_d is the density coefficient (K_d) that can be found in [25,26].

$K_{cb\ full}$ is calculated as a function of mean crop height and adjusted for local climate conditions. This is because the standard K_{cb} values refer to a typical sub-humid climate (with an average daily minimum RH of 45% and a wind speed of 2 ms^{-1}) [27], so the values of $K_{cb\ full}$ are adjusted using the following equation:

$$K_{cbfull} = K_{cb,\ h} + (0.04(u_2 - 2) - 0.004(RH_{min} - 45))\left(\frac{h}{3}\right)^{0.3} \tag{4}$$

where $K_{cb,\ h}$ is the $K_{cb\ mid}$ for full cover vegetation under sub-humid and calm wind conditions (RH_{min} 45% and $u_2 = 2\ \mathrm{ms}^{-1}$) that can be estimated ($K_{cb,\ h} = \min(1.0 + 0.1h,\ 1.20)$) [25], u_2 is the wind speed at 2-m height [ms^{-1}], and h is the mean maximum crop height [m].

Finally, K_{cb} is adjusted for available soil water to maintain crop ET at potential rate. K_s can be calculated by the following function [17,18]:

$$K_s = \frac{TAW - D_r}{TAW - RAW} \text{ when } D_r\ > RAW \tag{5}$$

where TAW and RAW are the total available water and readily available water in the root zone [mm], respectively, and Dr is the soil water depletion in the root zone relative to field capacity [mm] [17].

The last step in the estimation of actual crop coefficient ($K_{c\ act}$) is the calculation of K_e. This coefficient is highest when the topsoil is wet and decreases as the water content drops:

$$K_e = \min(K_r(K_{c\ max} - K_{cb}),\ f_{ew}K_{c\ max}) \tag{6}$$

where K_r is the dimensionless evaporation reduction coefficient dependent on the cumulative depth of water depleted from the topsoil and f_{ew} is the fraction of the soil surface wetted and exposed to solar radiation. Equations for calculating the K_r, $K_{c\ max}$, and f_{ew} are presented in the FAO56 report [17].

The simplified procedures used in the SIMDualKc [19] model have been used to calculate the last three components of the SWB (DP, CR, and RO). The SIMDualKc model calculates DP and CR using the procedure described by FAO-24 [27] and calculates RO using the curve number method [17].

2.3.2. Wet Bulb Calculation

As the crop used is drip irrigated, a series of equations have been developed to calculate the dimensions of the wet bulb of the soil to give more precise recommendations on the hours that the irrigation system should be in operation.

The model uses the soil properties to calculate the wet bulb dimensions and the soil water retention curve in order to estimate the maximum amount of water that can be applied without percolation and the TAW, respectively. Figure 7 gives an overview of the procedures followed.

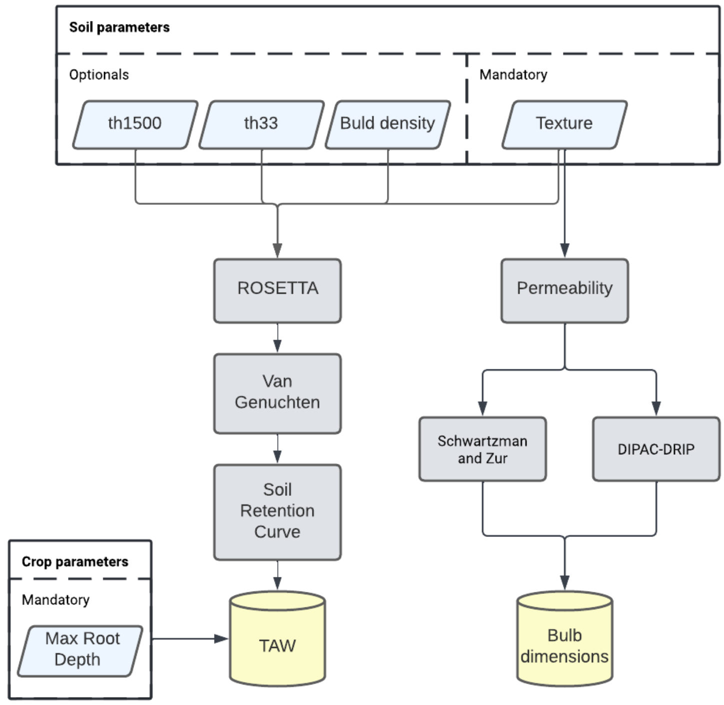

Figure 7. Flowchart of the procedures for calculating TAW and wet bulb dimensions. Where th1500 and th33 are the soil water content at soil matric potentials of −1500 and −33 kPa, respectively.

Calculation of Total Available Water

To facilitate the use of the irrigation model, the soil texture and the effective root depth are the only mandatory parameters to calculate TAW. To calculate the missing parameters, the Rosetta pedotransfer model, which predicts the hydraulic parameters of unsaturated soils, has been used. This model allows the estimation of van Genuchten water retention parameters and saturated hydraulic conductivity using limited (textural classes only) to more extended (texture, bulk density, and one or two water retention points) input data [28–30]. However, it is highly recommended to estimate all these soil parameters in the lab in order to use the extended version and get better results.

Using the parameters calculated by Rosetta, the van Genuchten equation can be used to calculate the soil water retention curve [31].

$$\theta(h) = \theta_r + \frac{\theta_s - \theta_r}{\left(1 + (\alpha h)^n\right)^{\left(1-\frac{1}{n}\right)}} \tag{7}$$

where $\theta(h)$ is the soil water content [m^3 m^{-3}] at a matric potential h [kPa], θ_s is the saturated water content, θ_r is the residual water content, α is related to the inverse of the air entry suction [cm^{-1}], and n is a measure of the pore-size distribution

Finally, with the calculated water retention curve, the soil TAW can be estimated as the difference between the volumetric soil water content at −33 kPa, which represents the field capacity (FC), and at −1500 kPa, which represents the permanent wilting point (PWP) [32] multiplied by the effective root depth.

Calculation of the Wet Bulb Dimensions

To calculate how long the irrigation system should run, it is necessary to know the dimensions of the wet bulb. There are several methods that can be used for this, which can be divided into empirical [33–35], numerical [36] and analytical models [37].

We have used two models to calculate bulb dimensions. The first is the Schwartzman and Zur model [33]:

$$w = 1.82\ V^{0.22}\left(\frac{K_{sat}}{Q}\right)^{-0.17} \tag{8}$$

$$z = 2.54\ V^{0.63}\left(\frac{K_{sat}}{Q}\right)^{0.45} \tag{9}$$

where w and z are the horizontal and vertical dimensions [m], V is the total amount of water in the soil [m^3], K_{sat} is the saturated hydraulic conductivity [ms^{-1}], and Q is the emitter discharge [m^{3}s^{-1}]

The second is the DIPAC-DRIP model. Equations (10) and (11) were created to estimate the horizontal and vertical dimensions of the wet bulb using a non-linear regression analysis.

$$w = 0.2476\ \Delta\theta^{-0.5626}\ V^{0.2686}\ Q^{-0.0028}\ {K_s}^{-0.0344} \tag{10}$$

$$z = 2.0336\ \Delta\theta^{-0.383}\ V^{0.365}\ Q^{-0.101}\ {K_s}^{0.195} \tag{11}$$

where $\Delta\theta$ is the average change of SWC ($\Delta\theta\ \approx \theta_s/2$, where θ_s is the soil water content at saturation).

The effective root depth parameter is used to calculate the dimensions of the wet bulb before percolation. With the volume of this wet bulb, the amount of water needed to bring it to the target moisture content can be estimated. To calculate this volume using the horizontal and vertical dimensions, the shape of the bulb has been assumed to be a semi-ellipse. This assumption is not entirely correct as the shape of the bulb is not exactly a semi-ellipse, and some studies have tried to calculate the shape more precisely [38]. However, it has been decided to use the semi-ellipse for simplification, and other studies will be taken into consideration for future improvements of the model.

The possible overlapping of the surface wetted by drippers in linear branches has been taken into account. Since the shape of the wet bulb is considered the same, the overlapped area can be calculated as the area of the intersection between two equal circles:

$$A_{c,overlaped} = 4\left(\frac{\pi R^2 cos^{-1}\frac{d}{2R}}{360} - \frac{R\left(\sqrt{d^2 - \frac{R^2}{4}}\right)}{4}\right) \tag{12}$$

where $A_{c,overlaped}$ is the overlapping area between the circles, R is the radius of the wet bulb, and d is the distance between the drippers. To translate this overlap area of the circles to the overlap area of the semi-ellipse, the following equation is used

$$A_{se,overlaped} = \frac{A_{c,overlaped}\left(\frac{z}{R}\right)}{2} \tag{13}$$

Finally, with the volume of the wet bulb, the overlapping area, and the soil humidity data, the total time that the irrigation system should work can be estimated as:

$$t = \frac{V_{se}\left(\frac{A_{se}}{A_{se,\ overlaped}}\right)\left(\frac{I}{1000\ z}\right)}{Q} \tag{14}$$

where I represents the irrigation needs [mm] calculated in the SWB, V_{se} is the volume of the semiellipse, and A_{se} is the area of the semiellipse.

2.3.3. Irrigation Strategies

The irrigation model uses four types of irrigation strategies: (1) Rainfed: when there is no irrigation; (2) Soil moisture target: define a maximum soil water depletion for each month of the year, which is particularly useful when scheduling deficit irrigation strategies;

(3) Predefined schedule: used if the irrigation is already defined and it is necessary to provide an irrigation time for each day of the simulation; (4) Constant depth: a strategy that attempts to keep soil water depletion within a specific range.

Other parameters, such as limiting the maximum irrigation time per day, the interval between irrigation days, or the days of the week on which irrigation can take place, are considered.

2.4. Server Implementation

Finally, once the design is finished, it is necessary to move the backend into production mode. This consists of deploying the application on a server so that it is operational 24 h a day. In this section, it is important to apply the corresponding security measures.

The development and deployment of the application was through Docker, a platform for the development of software containers. Among the advantages of using this technology are the possibility of running multiple servers on the same machine in isolation, scalability, and ease of deployment in the cloud.

As a server, we have used a Raspberry Pi 4 with 4 GB of RAM and a 64-bit Raspbian operating system.

3. Implementation

3.1. Device Implementation

The model described above was implemented in the IoT system that was evaluated on the olive farm during one irrigation season. Two communication nodes have been installed (Figure 8), one in each of the irrigation sectors or plots of olive trees. Each node consists of three soil humidity, electrical conductivity, and temperature probes, placed at 15, 30, and 45 cm soil depth, and a soil water potential sensor placed at 30 cm.

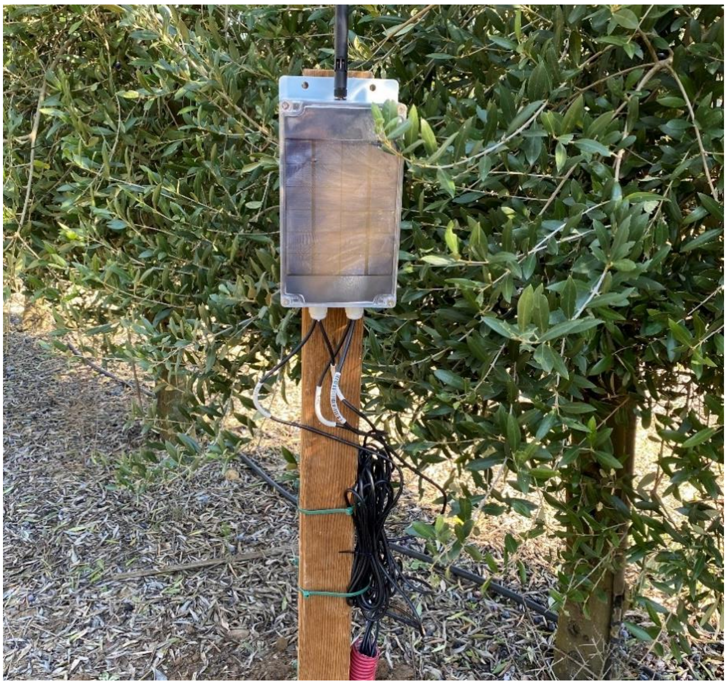

Figure 8. Communication node with connected soil moisture sensors.

The installation of the sensors is quite simple and intuitive. The communication node allows the connection of sensors with a 3.5 mm jack connector. As already mentioned, the architecture of the platform allows any type of sensor to be connected. To do this, when the user installs a new node, a window appears on the platform where the user must specify which sensors have been installed in each connector. Once this step is finished, the platform adapts to that sensor and displays its data.

3.2. User Interface

The application interface has been developed to be user-friendly and scalable. This interface allows access to the different entities within the farm in a quick and intuitive way (Figure 9).

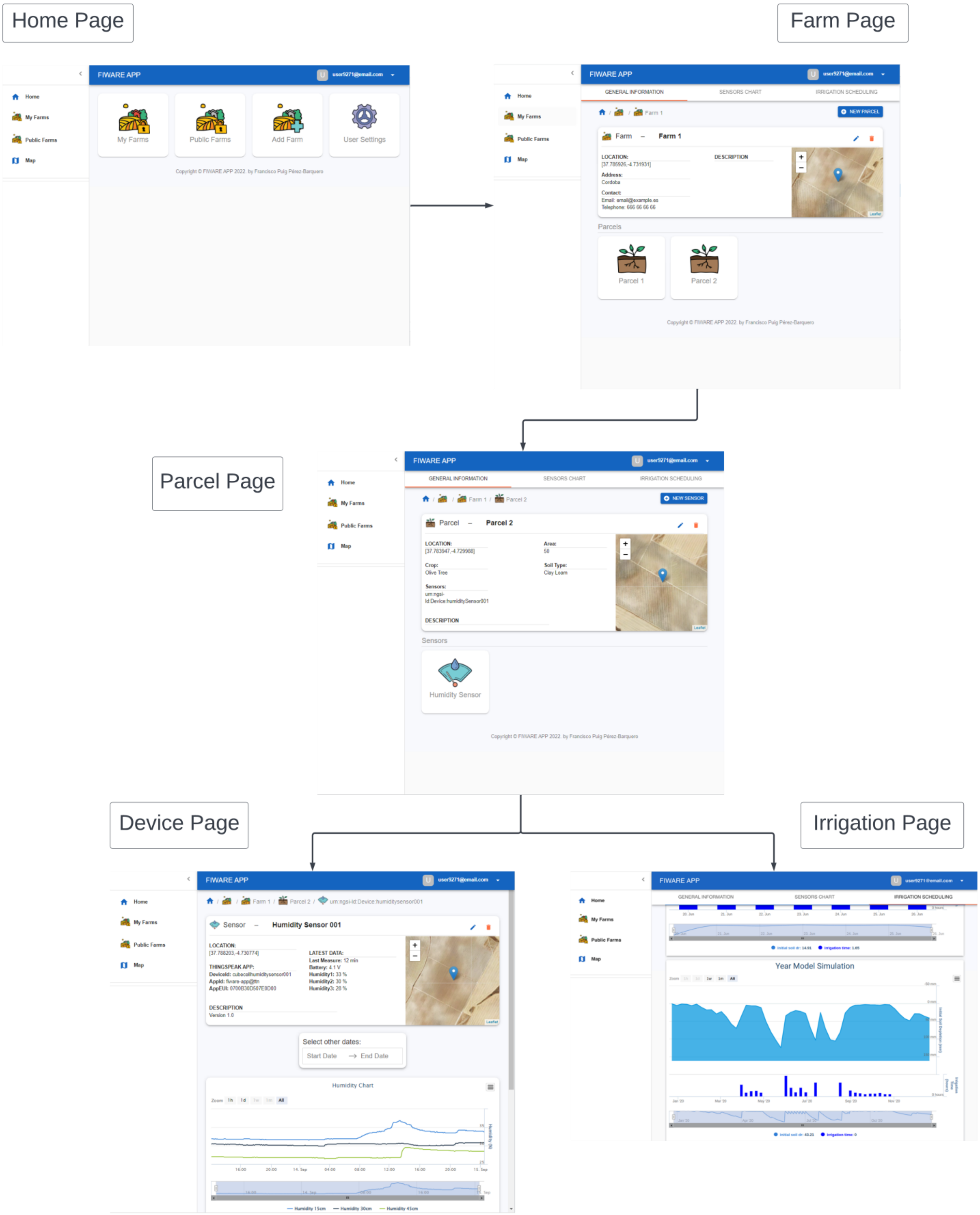

Figure 9. Main interface pages.

The home page allows access to the main menus of the application, such as the user settings or the farms. The farms page shows the general characteristics of the farm and the field plots that compose it; this entity stores weather data. The next page is the plots page; this is the most important one because it is where the irrigation is calculated. This page has all the crops and agronomic data, the irrigation schedules, and the sensors that are installed there. Finally, the interface has the page sensors, where all sensor characteristics and telemetry are displayed.

This structure allows for the separation of the different "Parcel" entities of a farm in order to use their characteristics as input for the agronomic models.

3.3. Scheduled Irrigation

As mentioned above, the model allows multiple irrigation strategies; the use of one or the other will depend on the soil characteristics, the crop, the limits of the irrigation system, and the user's management.

In this work, a regulated deficit irrigation (RDI) strategy has been chosen as the irrigation allocation is not sufficient to replenish 100% of the irrigation needs of the olive grove on the farm. This strategy maximizes water productivity by reducing applied water only during the least drought-sensitive crop phenological periods [39], in which the crop can suffer water stress without severely affecting crop yield.

For the two plots of olive trees monitored in this work, the RDI strategy evaluated in a high-density olive grove of the Arbequina variety located in the province of Seville was followed [40] to apply a total irrigation amount of 60% of irrigation needs (IN; Figure 10).

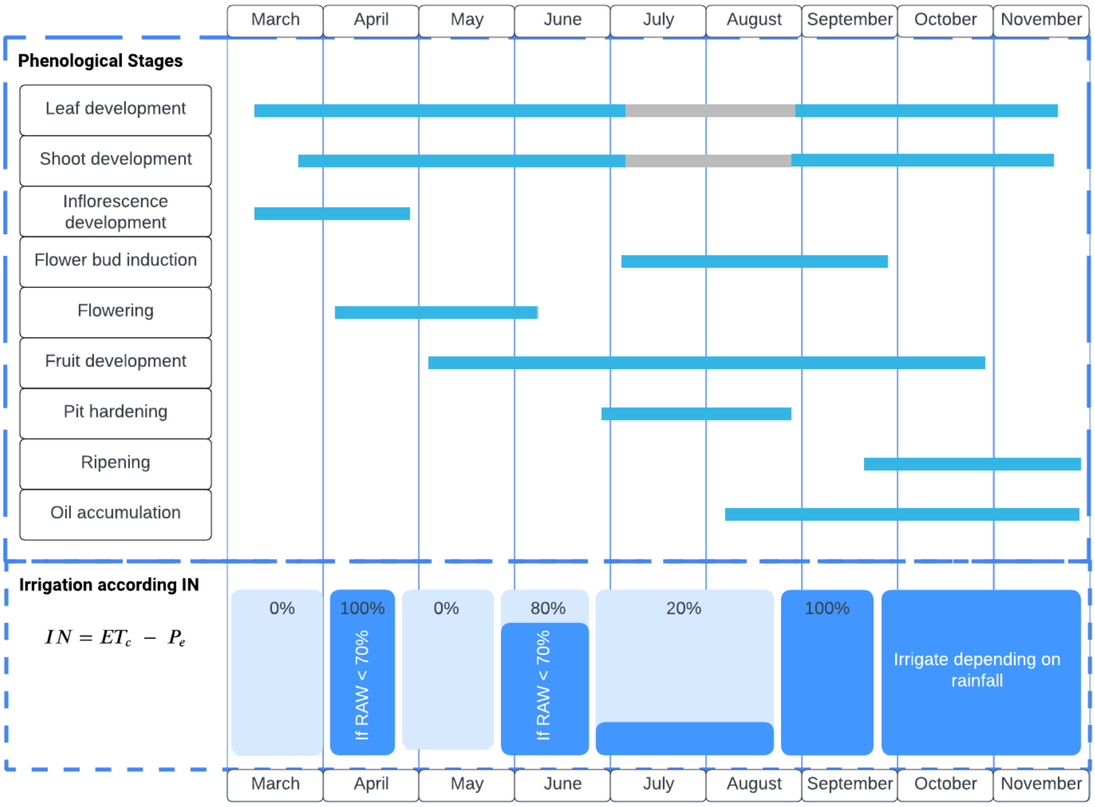

Figure 10. RDI strategy applied in a high-density olive grove to supply 60% of the crop's water needs [40]. Modified from [41].

To implement this irrigation scheduling, the strategy defined in the irrigation model as "soil moisture target" has been used. As mentioned above, with this strategy, the user can define a maximum soil water depletion for each month. Thus, the RDI strategy can be used in the water balance model. Following this strategy, the model performs two simulations. The first one is an annual simulation that is run using historical data from the last 20 years (Figure 11). This simulation shows how the water content in the soil changes during a normal year. The second simulation predicts the water needs of the olive crops in the following 7 days (Figure 12). For this, the soil water content is adjusted each day with the data provided by the soil moisture sensors, and the simulation of the water balance is performed using weather forecast data and the scheduled irrigation events.

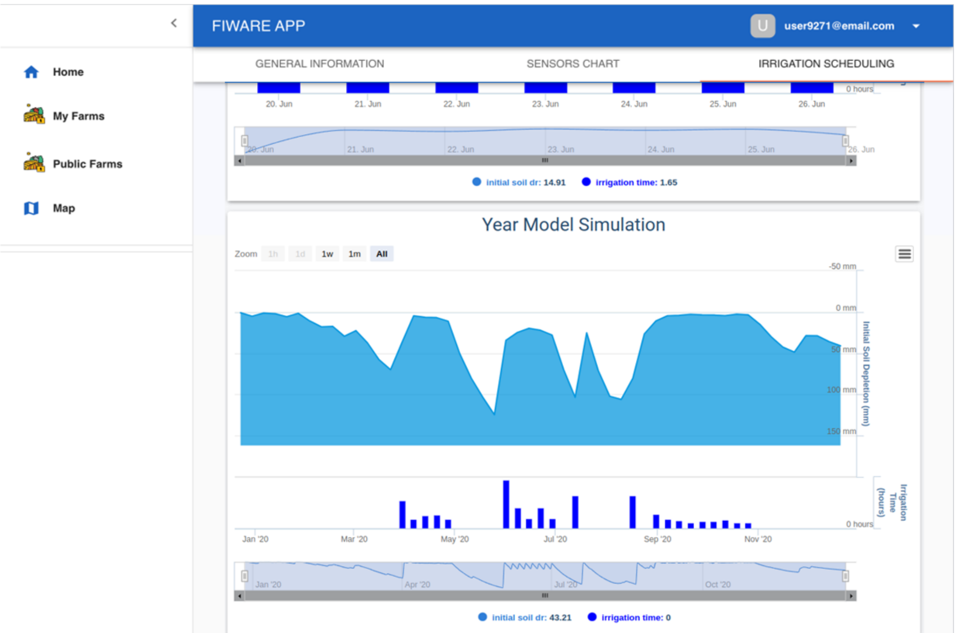

Figure 11. Annual simulation of the irrigation model showing water depletion in the root zone when using the RDI strategy defined in Figure 10 for a high-density olive grove.

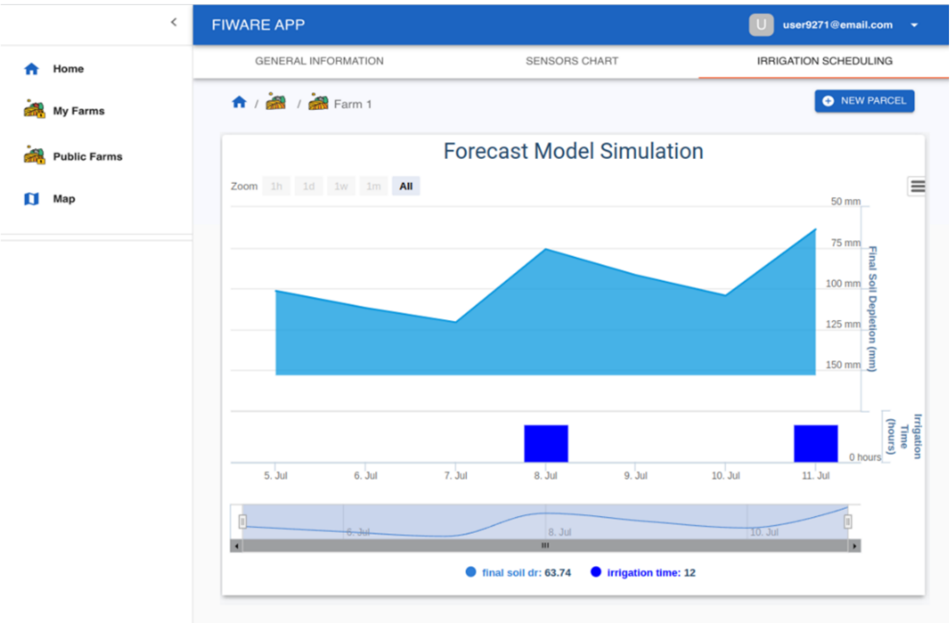

Figure 12. Simulation of the next seven days from the current day, showing water depletion in the root zone when using the strategy defined in Figure 10.

4. Discussion

While the use of sensors in agriculture is growing exponentially, the use of IoT platforms is also growing exponentially. The problem is that these platforms often have their own architecture, and the API is not standardized. This makes it difficult to connect between platforms and leads to their isolation. FIWARE, in particular its Linked-Data version, is developed with the aim of solving this problem and appears as a great alternative for IoT applications in agriculture.

That is why the most innovative part of this platform lies in its ease of sharing data between third-party platforms. Any platform developed with, or adapted to work with, the Orion-LD broker can be connected with little or no configuration. This is particularly beneficial in the public sector and in research, where data tends to become increasingly free, encouraging cooperation between researchers.

On the other hand, it is important that irrigation platforms use agronomic models that provide more information to the farmer. The reliability of these models depends on the homogeneity of the sector and the accuracy of the data obtained.

One of the great advantages of open platforms is that they are easily scalable and allow the integration of new sensors, which will be able to send real-time information to soil-water-plant-atmosphere models and considerably improve the accuracy of the estimates. For example, the integration of eddy covariance stations and sap flow sensors could help to improve the accuracy of the estimation of crop water requirements.

With all this, it will be possible to integrate large amounts of information, what we know as 'big data', in quite complex models while offering accurate recommendations to farmers in a user-friendly and easy-to-understand way.

In this particular case, before the implementation of this smart irrigation system, traditional irrigation management consisted of homogeneous irrigation of the olive farm, without making any distinction between the different sectors, despite the fact that the soil hydraulic characteristics and the olive varieties, as well as the planting date and tree density, were different. Moreover, the amount of irrigation depth applied was constant throughout the irrigation season, without taking into account the water requirements or the phenological stage of the olive tree. This was a problem, as there were periods of overwatering and others in which the needs of the olive trees were not supplied. By using the web application, which displayed the results of the sensors and the application of the irrigation model, the farmer improved his irrigation management, reducing water losses and avoiding tree water stress. It also facilitated the application of RDI strategies, something that is often not easy for the average farmer.

Thanks to the use of technology, this IoT platform for smart irrigation is able to know, in real time, the state of the soil and of the crop and is able to use already validated soil-water-plant-atmosphere models that give us current and future information on the behavior of crops, all in an automatic way. Although it is difficult to measure the efficiency of this system in comparison with the conventional ones, as the latter depend mainly on the knowledge and experience of the farmer, it is clear that the information provided by this smart irrigation system is very useful for making decisions regarding irrigation scheduling. Furthermore, for those farmers who are not familiar with agronomic models, this system allows them to use these models without having the necessary knowledge to understand them. On the other hand, for those more experienced farmers who already use this type of agronomic model, this system makes their work much easier as it allows them to automate irrigation scheduling.

5. Conclusions

The design of a complete open-source IoT system for smart irrigation systems is already completed. The system is a multilayer platform consisting of a series of energy-efficient smart devices connected to an IoT platform that is based on a microservices architecture. The platform uses the FIWARE framework alongside customized components and can be deployed using edge computing and/or cloud systems. This allows it to be adapted to the farmer's needs, reducing costs and increasing safety.

To complement the platform, an energy-efficient open-source datalogger has been designed. The datalogger supports a wide range of communications and protocols. The open-source availability of the data collected from the different sensors will facilitate the integration of the data into soil-water-plant-atmosphere models and their use as decision support systems for optimal irrigation management.

In addition, a soil-water-crop model has been developed to improve irrigation management. This model calculates the dimensions and characteristics of the wet bulb when there is a drip irrigation system and uses the dual Kc approach to calculate the crop water requirements. Two simulations are performed in the application: the first calculates the soil water balance for a normal year, and the second calculates the water balance for the next seven days.

Author Contributions: Conceptualization, M.A.S., J.A.R.D. and F.P.; methodology, F.P., M.A.S. and J.A.R.D.; software, F.P.; validation, M.A.S., J.A.R.D. and F.P.; investigation, F.P.; writing—original draft preparation, F.P.; writing—review and editing, J.A.R.D. and M.A.S.; visualization, M.A.S., J.A.R.D. and F.P.; supervision, J.A.R.D. and M.A.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the María de Maeztu Unit of Excellence of the Department of Agronomy at the University of Cordoba.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. García Morillo, J. *Hacia El Riego de Precisión En El Cultivo de Fresa En El Entorno de Doñana*; Universidad de Córdoba; UCOPress: Córdoba, Spain, 2015.
2. Zal, N.; Wolters, H.; Psomas, A.; Anzaldua, G.; Bariamis, G.; Rouillard, J.; Birk, S. *Water Resources across Europe—Confronting Water Stress: An Updated Assessment*; European Environment Agency: Copenhagen, Denmark, 2021.
3. Muñoz, M.; Gil, J.D.; Roca, L.; Rodríguez, F.; Berenguel, M. An IoT Architecture for Water Resource Management in Agroindustrial Environments: A Case Study in Almería (Spain). *Sensors* **2020**, *20*, 596. [CrossRef] [PubMed]
4. Kamienski, C.; Soininen, J.-P.; Taumberger, M.; Dantas, R.; Toscano, A.; Salmon Cinotti, T.; Filev Maia, R.; Torre Neto, A. Smart Water Management Platform: IoT-Based Precision Irrigation for Agriculture. *Sensors* **2019**, *19*, 276. [CrossRef] [PubMed]
5. Zamora-Izquierdo, M.A.; Santa, J.; Martínez, J.A.; Martínez, V.; Skarmeta, A.F. Smart Farming IoT Platform Based on Edge and Cloud Computing. *Biosyst. Eng.* **2019**, *177*, 4–17. [CrossRef]
6. Alonso, R.S.; Sittón-Candanedo, I.; García, Ó.; Prieto, J.; Rodríguez-González, S. An Intelligent Edge-IoT Platform for Monitoring Livestock and Crops in a Dairy Farming Scenario. *Ad Hoc Netw.* **2020**, *98*, 102047. [CrossRef]
7. FIWARE Open Source Platform for the Smart Digital Future. Available online: https://www.fiware.org/about-us/ (accessed on 9 November 2022).
8. López-Morales, J.A.; Martínez, J.A.; Skarmeta, A.F. Digital Transformation of Agriculture through the Use of an Interoperable Platform. *Sensors* **2020**, *20*, 1153. [CrossRef] [PubMed]
9. Trilles, S.; González-Pérez, A.; Huerta, J. An IoT Platform Based on Microservices and Serverless Paradigms for Smart Farming Purposes. *Sensors* **2020**, *20*, 2418. [CrossRef]
10. Alcaide Zaragoza, C.; González Perea, R.; Fernández García, I.; Camacho Poyato, E.; Rodríguez Díaz, J.A. Open Source Application for Optimum Irrigation and Fertilization Using Reclaimed Water in Olive Orchards. *Comput. Electron. Agric.* **2020**, *173*, 105407. [CrossRef]
11. Zaragoza, C.A.; Fernández García, I.; Perea, R.G.; Poyato, E.C.; Antonio, J.; Díaz, R. REUTIVAR: Model for Precision Fertigation Scheduling for Olive Orchards Using Reclaimed Water. *Water* **2019**, *11*, 2632. [CrossRef]
12. Allen, R.G.; Pereira, L.S.; Smith, M.; Raes, D.; Wright, J.L. FAO-56 Dual Crop Coefficient Method for Estimating Evaporation from Soil and Application Extensions. *J. Irrig. Drain. Eng.* **2005**, *131*, 2–13. [CrossRef]
13. Sethi, P.; Sarangi, S.R. Internet of Things: Architectures, Protocols, and Applications. *J. Electr. Comput. Eng.* **2017**, *2017*, 9324035. [CrossRef]
14. FIWARE: Orion Context Broker. Available online: https://fiware-orion.readthedocs.io/en/master/ (accessed on 20 July 2021).
15. Context Information Management (CIM); NGSI-LD API. Available online: https://www.etsi.org/deliver/etsi_gs/CIM/001_099/009/01.04.01_60/gs_cim009v010401p.pdf (accessed on 22 July 2021).
16. Schema.Org. Available online: https://schema.org/ (accessed on 22 July 2021).
17. Allen, R.G.; Pereira, L.S.; Raes, D.; Smith, M. *Crop Evapotranspiration—Guidelines for Computing Crop Water Requirements*; FAO Irrigation and Drainage Paper No. 56; FAO: Rome, Italy, 1998.
18. Pereira, L.S.; Paredes, P.; Hunsaker, D.J.; López-Urrea, R.; Mohammadi Shad, Z. Standard Single and Basal Crop Coefficients for Field Crops. Updates and Advances to the FAO56 Crop Water Requirements Method. *Agric. Water Manag.* **2021**, *243*, 106466. [CrossRef]

19. Rosa, R.D.; Paredes, P.; Rodrigues, G.C.; Alves, I.; Fernando, R.M.; Pereira, L.S.; Allen, R.G. Implementing the Dual Crop Coefficient Approach in Interactive Software. 1. Background and Computational Strategy. *Agric. Water Manag.* **2012**, *103*, 8–24. [CrossRef]
20. Paço, T.A.; Pôças, I.; Cunha, M.; Silvestre, J.C.; Santos, F.L.; Paredes, P.; Pereira, L.S. Evapotranspiration and Crop Coefficients for a Super Intensive Olive Orchard. An Application of SIMDualKc and METRIC Models Using Ground and Satellite Observations. *J. Hydrol.* **2014**, *519*, 2067–2080. [CrossRef]
21. Paço, T.A.; Paredes, P.; Pereira, L.S.; Silvestre, J.; Santos, F.L. Crop Coefficients and Transpiration of a Super Intensive Arbequina Olive Orchard Using the Dual Kc Approach and the Kcb Computation with the Fraction of Ground Cover and Height. *Water* **2019**, *11*, 383. [CrossRef]
22. Puig-Sirera, À.; Rallo, G.; Paredes, P.; Paço, T.A.; Minacapilli, M.; Provenzano, G.; Pereira, L.S. Transpiration and Water Use of an Irrigated Traditional Olive Grove with Sap-Flow Observations and the Fao56 Dual Crop Coefficient Approach. *Water* **2021**, *13*, 2466. [CrossRef]
23. Paço, T.A.; Ferreira, M.I.; Rosa, R.D.; Paredes, P.; Rodrigues, G.C.; Conceição, N.; Pacheco, C.A.; Pereira, L.S. The Dual Crop Coefficient Approach Using a Density Factor to Simulate the Evapotranspiration of a Peach Orchard: SIMDualKc Model versus Eddy Covariance Measurements. *Irrig. Sci.* **2012**, *30*, 115–126. [CrossRef]
24. Fandiño, M.; Cancela, J.J.; Rey, B.J.; Martínez, E.M.; Rosa, R.G.; Pereira, L.S. Using the Dual-K c Approach to Model Evapotranspiration of Albariño Vineyards (Vitis Vinifera L. Cv. Albariño) with Consideration of Active Ground Cover. *Agric. Water Manag.* **2012**, *112*, 75–87. [CrossRef]
25. Allen, R.G.; Pereira, L.S. Estimating Crop Coefficients from Fraction of Ground Cover and Height. *Irrig. Sci.* **2009**, *28*, 17–34. [CrossRef]
26. Pereira, L.S.; Paredes, P.; Melton, F.; Johnson, L.; Mota, M.; Wang, T. Prediction of Crop Coefficients from Fraction of Ground Cover and Height: Practical Application to Vegetable, Field and Fruit Crops with Focus on Parameterization. *Agric. Water Manag.* **2021**, *252*, 106663. [CrossRef]
27. Rallo, G.; Paço, T.A.; Paredes, P.; Puig-Sirera, A.; Massai, R.; Provenzano, G.; Pereira, L.S. Updated Single and Dual Crop Coefficients for Tree and Vine Fruit Crops. *Agric. Water Manag.* **2021**, *250*, 106645. [CrossRef]
28. Schaap, M.G.; Leij, F.J.; Van Genuchten, M.T. Rosetta: A Computer Program for Estimating Soil Hydraulic Parameters with Hierarchical Pedotransfer Functions. *J. Hydrol.* **2001**, *251*, 163–176. [CrossRef]
29. Schaap, M.G.; Nemes, A.; Genuchten, M.T. Comparison of Models for Indirect Estimation of Water Retention and Available Water in Surface Soils. *Vadose Zone J.* **2004**, *3*, 1455–1463. [CrossRef]
30. Zhang, Y.; Schaap, M.G. Weighted Recalibration of the Rosetta Pedotransfer Model with Improved Estimates of Hydraulic Parameter Distributions and Summary Statistics (Rosetta3). *J. Hydrol.* **2017**, *547*, 39–53. [CrossRef]
31. van Genuchten, M.T. A Closed-Form Equation for Predicting the Hydraulic Conductivity of Unsaturated Soils. *Soil Sci. Soc. Am. J.* **1980**, *44*, 892–898. [CrossRef]
32. Salter, P.J.; Berry, G.; Williams, J.B. The influence of texture on the moisture characteristics of soils: III. Quantitative relationships between particle size, composition, and available-water capacity. *J. Soil Sci.* **1966**, *17*, 93–98. [CrossRef]
33. Schwartzman, M.; Zur, B. Emitter Spacing and Geometry of Wetted Soil Volume. *J. Irrig. Drain. Eng.* **1986**, *112*, 242–253. [CrossRef]
34. Zur, B. Wetted Soil Volume as a Design Objective in Trickle Irrigation. *Irrig. Sci.* **1996**, *16*, 101–105. [CrossRef]
35. Amin, M.S.M.; Ekhmaj, A.I.M. DIPAC-Drip Irrigation Water Distribution Pattern Calculator. In Proceedings of the 7th International Micro Irrigation Congress, PWTC, Kuala Lumpur, Malaysia, 10–16 September 2006.
36. Lazarovitch, N.; Poulton, M.; Furman, A.; Warrick, A.W. Water Distribution under Trickle Irrigation Predicted Using Artificial Neural Networks. *J. Eng. Math.* **2009**, *64*, 207–218. [CrossRef]
37. Thorburn, P.J.; Cook, F.J.; Bristow, K.L. Soil-Dependent Wetting from Trickle Emitters: Implications for System Design and Management. *Irrig. Sci.* **2003**, *22*, 121–127. [CrossRef]
38. Kilic, M. A New Analytical Method for Estimating the 3D Volumetric Wetting Pattern under Drip Irrigation System. *Agric. Water Manag.* **2020**, *228*, 105898. [CrossRef]
39. Fereres, E.; Soriano, M.A. Deficit Irrigation for Reducing Agricultural Water Use. *J. Exp. Bot.* **2007**, *58*, 147–159. [CrossRef] [PubMed]
40. Fernández, J.E.; Perez-Martin, A.; Torres-Ruiz, J.M.; Cuevas, M.V.; Rodriguez-Dominguez, C.M.; Elsayed-Farag, S.; Morales-Sillero, A.; García, J.M.; Hernandez-Santana, V.; Diaz-Espejo, A.; et al. A Regulated Deficit Irrigation Strategy for Hedgerow Olive Orchards with High Plant Density. *Plant Soil* **2013**, *372*, 279–295. [CrossRef]
41. Gucci, R.; Fereres, E.; Goldhamer, D.A. Olive: Crop Yield Response to Water. In *Irrigation and Drainage Paper No. 66*, 2nd ed.; Steduto, P., Hsiao, T.C., Fereres, E., Raes, D., Eds.; FAO: Rome, Italy, 2012; pp. 300–313.