



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería
Informática**

ARBUBU



Presentado por Félix Movilla Alonso
en Universidad de Burgos — 7 de enero
de 2020

Tutor: Pedro Renedo Fernández y Antonio
Jesús Canepa Oneto

Índice general

Índice general	I
Índice de figuras	III
Índice de tablas	IV
Apéndice A Plan de Proyecto Software	1
A.1. Introducción	1
A.2. Planificación temporal	1
A.3. Estudio de viabilidad	1
Apéndice B Especificación de Requisitos	3
B.1. Introducción	3
B.2. Objetivos	3
B.3. Catalogo de requisitos	4
B.4. Especificación de requisitos	6
Apéndice C Especificación de diseño	13
C.1. Introducción	13
C.2. Diseño de datos	13
C.3. Diseño procedimental	18
C.4. Diseño arquitectónico	20
Apéndice D Documentación técnica de programación	23
D.1. Introducción	23
D.2. Estructura de directorios	23
D.3. Manual del programador	23

D.4. Compilación, instalación y ejecución del proyecto	23
D.5. Pruebas del sistema	23
Apéndice E Documentación de usuario	25
E.1. Introducción	25
E.2. Requisitos de usuarios	25
E.3. Instalación	25
E.4. Manual del usuario	25
Bibliografía	27

Índice de figuras

B.1. Diagrama de Casos de Uso	6
C.1. Tabla Familia	13
C.2. Tabla Genero	14
C.3. Tabla Especie	15
C.4. Tabla Individuos	16
C.5. Tabla Usuario	17
C.6. Diagrama de Navegabilidad	20

Índice de tablas

B.1. CU-1 Cargar datos.	7
B.2. CU-2 Exportar datos.	8
B.3. CU-3 Visualizar datos.	9
B.4. CU-4 Controlar usuarios.	10
B.5. CU-5 Buscar datos específicos.	11
B.6. CU-6 Visualizado en varios dispositivos.	12

Apéndice A

Plan de Proyecto Software

A.1. Introducción

A.2. Planificación temporal

A.3. Estudio de viabilidad

Viabilidad económica

Viabilidad legal

Apéndice B

Especificación de Requisitos

B.1. Introducción

Este anexo se encarga de mostrar los objetivos generales de la aplicación web, además de detallar los requisitos funcionales y no funcionales.

B.2. Objetivos

El principal objetivo del proyecto es realizar un diseño web, en el cual se puedan ver los árboles singulares de las zonas universitarias de Burgos, con sus principales características.

A través de un mapa podremos ver donde están ubicados los árboles.

Objetivos Generales

- Observar en un mapa los árboles singulares de las zonas universitarias de Burgos y ver sus características.
- Filtrar a través de la familia, género, especie y ubicación los distintos árboles.
- Loguearnos como usuario y ser capaces de introducir y descargar datos de los árboles.
- Realizar una primera toma de contacto con la búsqueda de árboles, que en una futura mejora no solo busquemos arboles, es decir, que seamos capaces de buscar monumentos, lugares importantes ...

- Poder compartir y dar me gusta a la página de facebook de UbuVerde e interactuar con ellos a través de twitter.
- Estar en constante contacto con UbuVerde a través de 'Acerca de Nosotros' que es un enlace a la página de UbuVerde.

Objetivos Técnicos

- Crear la Base de Datos para guardar la información de las familias, géneros, especies e individuos con una estructura adecuada de tablas y campos.
- Ser capaz de introducir nuevos individuos en la base de datos Sqlite3.
- Ser capaz de descargar información de la base de datos Sqlite3.
- Plasmar en el mapa esos datos introducidos en la base de datos.
- Desarrollar el diseño web con Django quedando plasmado con una interfaz gráfica amena y agradable para el usuario.
- Guardar en un repositorio de GitHub los cambios que hemos ido realizando.

B.3. Catalogo de requisitos

Requisitos funcionales

- **RF-1 Cargar datos:** Los usuarios deben ser capaces de introducir datos en la base de datos y poder visualizarlos después.
- **RF-2 Exportar datos:** Los usuarios deben ser capaces de exportar datos de la base de datos, para poder guardar la información de los árboles sin necesidad de visitar la página.
- **RF-3 Visualizar datos:** Los usuarios deben ser capaces de ver en un mapa los distintos árboles situados en las zonas universitarias.
 - **RF-3.1:** Se podrán filtrar los árboles a través de la familia, género, especie y ubicación de los distintos árboles.

- **RF-4 Control de usuarios:** La aplicación debe ser capaz de tener controlado en todo momento al usuario logueado.

RF-4.1 Registro: Los usuarios pueden registrarse para tener acceso a la descarga de los datos, así como su subida.

RF-4.2 Iniciar sesión: Los usuarios pueden iniciar sesión con una cuenta previamente registrada.

RF-4.3 Cierre de sesión: Los usuarios una vez terminada su visita en la página pueden cerrar sesión.

- **RF-5 Búsqueda de datos:** Los usuarios deben ser capaces de buscar datos de un árbol específico y ver donde está situado.
- **RF-6 Aspecto visual:** Los usuarios deben ser capaces de visualizar las distintas pantallas de la aplicación en varios dispositivos con distintos tamaños y que el aspecto visual siga mostrándose de igual calidad.

Requisitos no funcionales

- **RNF-1 Usabilidad:** La aplicación web debe ser intuitiva y de fácil manejo para el usuario.
- **RNF-2 Mantenibilidad:** La aplicación web debe ser capaz de añadir nuevos datos.
- **RNF-3 Compatibilidad:** La aplicación web debe poder visualizarse en los principales navegadores, así como en los dispositivos móviles.
- **RNF-4 Rendimiento:** La aplicación web debe cargar los datos y mapas con una velocidad adecuada.
- **RNF-5 Responsividad:** La aplicación web debe poder visualizarse sin perder calidad y adaptarse al tamaño en los principales navegadores, así como en los dispositivos móviles.
- **RNF-6 Escalabilidad:** A un mayor incremento de recursos la aplicación web debe ser capaz de incrementar en consecuencia su rendimiento.
- **RNF-7 Desplegabilidad:** La aplicación web debe ser capaz de integrarse en un servidor sin ningún problema.

B.4. Especificación de requisitos

Este apartado será el encargado de mostrar los diagramas de casos de uso basados en los requisitos funcionales del proyecto, para ello se describirán tanto en forma de tabla como de diagrama. Ver figura B.1

Diagrama de Casos de Uso

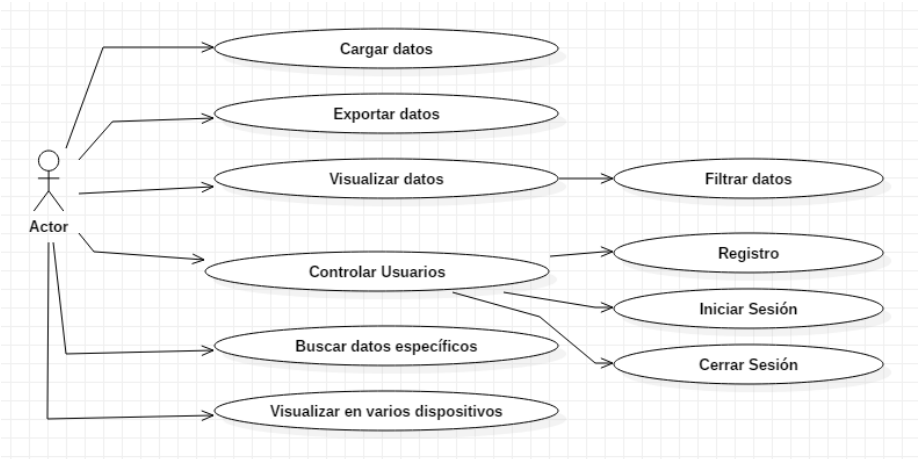


Figura B.1: Diagrama de Casos de Uso

Descripción de Casos de Uso

A continuación se mostrará una tabla para cada uno de los casos de uso.

CU-1	Cargar datos
Versión	1.0
Autor	Félix Movilla Alonso
Requisitos asociados	RF-1
Descripción	Carga datos en la base de datos para poder visualizarlos después.
Precondiciones	El usuario abre el navegador, carga la página de la aplicación y se registre.
Acciones	<ol style="list-style-type: none">1. El usuario inicie sesión.2. El usuario introduce los datos del árbol que desea añadir.
Postcondiciones	La información introducida es supervisada por el administrador y si es correcta se carga en la base de datos.
Excepciones	<ul style="list-style-type: none">■ La información introducida no es correcta.■ La información introducida es correcta pero no se puede catalogar como árbol singular.
Importancia	Alta

Tabla B.1: CU-1 Cargar datos.

CU-2	Exportar datos
Versión	1.0
Autor	Félix Movilla Alonso
Requisitos asociados	RF-2
Descripción	Exportar datos desde base de datos para poder guardar esa información.
Precondiciones	El usuario abre el navegador, carga la página de la aplicación y se registre.
Acciones	<ol style="list-style-type: none"> 1. El usuario inicie sesión. 2. El usuario exporta los datos de los árboles.
Postcondiciones	Si el usuario está registrado podrá descargar los datos.
Excepciones	<ul style="list-style-type: none"> ▪ Si el usuario no está registrado no le dejará descargar los datos.
Importancia	Media

Tabla B.2: CU-2 Exportar datos.

CU-3	Visualizar datos
Versión	1.0
Autor	Félix Movilla Alonso
Requisitos asociados	RF-3, RF-3.1
Descripción	Visualizar en un mapa los árboles situados en las zonas universitarias.
Precondiciones	El usuario abre el navegador, carga la página de la aplicación y que haya árboles introducidos.
Acciones	<ol style="list-style-type: none"> 1. Inicia la aplicación web y podrá ver los árboles situados en las zonas universitarias. 2. Se podrán filtrar los árboles a través de la familia, género, especie y ubicación universitaria de los distintos árboles.
Postcondiciones	Se muestra los datos del árbol seleccionado, así como la ubicación en el mapa de dicho árbol.
Excepciones	<ul style="list-style-type: none"> ■ Error si los datos seleccionados no coinciden con ningún dato de la base de datos.
Importancia	Alta

Tabla B.3: CU-3 Visualizar datos.

CU-4	Controlar usuarios
Versión	1.0
Autor	Félix Movilla Alonso
Requisitos asociados	RF-4, RF-4.1, RF-4.2, RF-4.3
Descripción	Controlar a los usuarios que desean visitar la página web.
Precondiciones	El usuario abre el navegador y carga la página de la aplicación.
Acciones	<ol style="list-style-type: none"> 1. El usuario pulsa el botón de registrarse. 2. El usuario introduce los datos para registrarse. 3. El usuario pulsa el botón de iniciar sesión. 4. El usuario introduce los datos para iniciar sesión. 5. El usuario pulsa el botón de cerrar sesión.
Postcondiciones	<p>Redirecciona a la página con la sesión iniciada.</p> <p>Redirecciona a la página con la sesión cerrada.</p>
Excepciones	<ul style="list-style-type: none"> ■ Error si los datos introducidos al registrarse no son correctos o ya existen. ■ Error si al iniciar sesión los datos introducidos no existen.
Importancia	Alta

Tabla B.4: CU-4 Controlar usuarios.

CU-5	Buscar datos específicos
Versión	1.0
Autor	Félix Movilla Alonso
Requisitos asociados	RF-5
Descripción	Buscar un árbol específico para ver donde está situado.
Precondiciones	El usuario abre el navegador y carga la página de la aplicación.
Acciones	<ol style="list-style-type: none"> 1. El usuario introduce la familia, género, especie o ubicación del árbol seleccionado y lo selecciona.
Postcondiciones	Se muestra en el mapa el árbol seleccionado y sus características.
Excepciones	<ul style="list-style-type: none"> ■ Error si los datos introducidos del árbol no existen.
Importancia	Media

Tabla B.5: CU-5 Buscar datos específicos.

CU-6	Visualizado en varios dispositivos
Versión	1.0
Autor	Félix Movilla Alonso
Requisitos asociados	RF-6
Descripción	Ver en distintos dispositivos la página web sin perder calidad.
Precondiciones	El usuario abre el navegador y carga la página de la aplicación.
Acciones	<ol style="list-style-type: none"> 1. El usuario abre la página web con el móvil, tablet, distintos navegadores de ordenador.
Postcondiciones	La página web se ve correctamente.
Excepciones	<ul style="list-style-type: none"> ■ Error si la página web no se ve correctamente o pierde calidad.
Importancia	Media

 Tabla B.6: CU-6 Visualizado en varios dispositivos.

Apéndice C

Especificación de diseño

C.1. Introducción

En este apartado explicaremos el diseño que ha dado lugar a la aplicación, para ello el anexo está dividido en tres secciones: el diseño de datos, el diseño procedimental y el diseño arquitectónico.

C.2. Diseño de datos

Para el almacenaje de los datos de la aplicación he utilizado una base de datos sqlite3, la cual está compuesta por seis tablas o modelos.

familia		CREATE TABLE "principal_familia" ("idFamilia" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "nombreFamilia" varchar(30) NOT NULL UNIQUE)
idFamilia	integer	"idFamilia" integer NOT NULL PRIMARY KEY AUTOINCREMENT
nombreFamilia	varchar(30)	"nombreFamilia" varchar(30) NOT NULL UNIQUE

Figura C.1: Tabla Familia

- **FAMILIA:** En este modelo se va a guardar la información relacionada con las familias de árboles existentes en las universidades de Burgos, está compuesta por dos campos que son:

idFamilia: Es la clave primaria de la tabla y contiene un identificador único para cada familia.

nombreFamilia: Contiene el nombre de la familia, es un campo de tipo texto, tiene que ser único, no se puede dejar en blanco y con un tamaño máximo de 30 caracteres.

genero		CREATE TABLE "principal_genero" ("idGenero" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "nombreGenero" varchar(30) NOT NULL UNIQUE, "familia_id" integer NOT NULL REFERENCES "principal_familia" ("idFamilia") DEFERRABLE INITIALLY DEFERRED)
idGenero	integer	"idGenero" integer NOT NULL PRIMARY KEY AUTOINCREMENT
nombreGenero	varchar(30)	"nombreGenero" varchar(30) NOT NULL UNIQUE
familia	integer	"familia_id" integer NOT NULL

Figura C.2: Tabla Genero

- **GENERO:** En este modelo se va a guardar la información relacionada con los géneros de árboles existentes en las universidades de Burgos, está compuesta por tres campos que son:

idGenero: Es la clave primaria de la tabla y contiene un identificador único para cada género.

nombreFamilia: Contiene el nombre del género, es un campo de tipo texto, tiene que ser único, no se puede dejar en blanco y con un tamaño máximo de 30 caracteres.

familia: Contiene el nombre de la familia, es una FK.

especie		CREATE TABLE "principal_especie" ("idEspecie" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "nombreCientificoEspecie" varchar(50) NOT NULL, "autoctona" bool NOT NULL, "descripcion" text NOT NULL, "ecologia" text NOT NULL, "genero_id" integer NOT NULL REFERENCES "principal_genero" ("idGenero") DEFERRABLE INITIALLY DEFERRED, "nombreComunEspecie" varchar(50) NOT NULL)
idEspecie	integer	"idEspecie" integer NOT NULL PRIMARY KEY AUTOINCREMENT
nombreCientificoEspecie	varchar(50)	"nombreCientificoEspecie" varchar(50) NOT NULL
autoctona	bool	"autoctona" bool NOT NULL
descripcion	text	"descripcion" text NOT NULL
ecologia	text	"ecologia" text NOT NULL
genero	integer	"genero_id" integer NOT NULL
nombreComunEspecie	varchar(50)	"nombreComunEspecie" varchar(50) NOT NULL

Figura C.3: Tabla Especie

- **ESPECIE:** En este modelo se va a guardar la información relacionada con las especies de árboles existentes en las universidades de Burgos, está compuesta por siete campos que son:

idEspecie: Es la clave primaria de la tabla y contiene un identificador único para cada especie.

nombreCientificoEspecie: Contiene el nombre científico de la especie, es un campo de tipo texto, no se puede dejar en blanco y con un tamaño máximo de 50 caracteres.

nombreComunEspecie: Contiene el nombre común de la especie, es un campo de tipo texto, no se puede dejar en blanco y con un tamaño máximo de 50 caracteres.

genero: Contiene el nombre del género, es una FK.

autoctona: Contiene si la especie es autóctona, es un campo de tipo booleano y no se puede dejar en blanco.

descripcion: Contiene una pequeña descripción de la especie, es un campo de tipo texto y no se puede dejar en blanco.

ecologia: Contiene la ecología de la especie, es un campo de tipo texto y no se puede dejar en blanco.

individuos		<pre>CREATE TABLE "principal_individuos" ("idIndividuo" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "nombreComun" varchar(30) NOT NULL, "motivoSingular" varchar(50) NOT NULL, "explicacionMotivoSingular" text NOT NULL, "fotoArbol" varchar(100) NOT NULL, "fotoHojas" varchar(100) NOT NULL, "fotoTronco" varchar(100) NOT NULL, "fotoFrutos" varchar(100) NOT NULL, "altura" decimal NOT NULL, "perimetro" decimal NOT NULL, "especie_id" integer NOT NULL REFERENCES "principal_especie" ("idEspecie") DEFERRABLE INITIALLY DEFERRED, "latitud" decimal NOT NULL, "longitud" decimal NOT NULL, "ubicacion" varchar(50) NOT NULL)</pre>
idIndividuo	integer	"idIndividuo" integer NOT NULL PRIMARY KEY AUTOINCREMENT
nombreComun	varchar(30)	"nombreComun" varchar(30) NOT NULL
motivoSingular	varchar(50)	"motivoSingular" varchar(50) NOT NULL
explicacionMotivoSingular	text	"explicacionMotivoSingular" text NOT NULL
fotoArbol	varchar(100)	"fotoArbol" varchar(100) NOT NULL
fotoHojas	varchar(100)	"fotoHojas" varchar(100) NOT NULL
fotoTronco	varchar(100)	"fotoTronco" varchar(100) NOT NULL
fotoFrutos	varchar(100)	"fotoFrutos" varchar(100) NOT NULL
altura	decimal	"altura" decimal NOT NULL
perimetro	decimal	"perimetro" decimal NOT NULL
especie	integer	"especie_id" integer NOT NULL
latitud	decimal	"latitud" decimal NOT NULL
longitud	decimal	"longitud" decimal NOT NULL
ubicacion	varchar(50)	"ubicacion" varchar(50) NOT NULL

Figura C.4: Tabla Individuos

- **INDIVIDUO:** En este modelo se va a guardar la información relacionada con cada uno de árboles existentes en las universidades de Burgos, está compuesta por catorce campos que son:

idIndividuo: Es la clave primaria de la tabla y contiene un identificador único para cada individuo.

nombreComun: Contiene el nombre común del árbol, es un campo de tipo texto, no se puede dejar en blanco y con un tamaño máximo de 30 caracteres.

especie: Contiene el nombre de la especie, es una FK.

motivoSingular: Contiene un motivo de por qué el árbol es singular, es un campo de tipo texto, no se puede dejar en blanco y con un tamaño máximo de 50 caracteres.

explicacionMotivoSingular: Contiene una explicación más detallada de porque el árbol es singular, es un campo de tipo texto y no se puede dejar en blanco.

latitud: Contiene la latitud del árbol, es un campo de tipo decimal y no se puede dejar en blanco.

longitud: Contiene la longitud del árbol, es un campo de tipo decimal y no se puede dejar en blanco.

fotoArbol: Contiene la foto del árbol, es un campo de tipo imagen y si se puede dejar en blanco.

fotoHojas: Contiene la foto de las hojas del árbol, es un campo de tipo imagen y si se puede dejar en blanco.

fotoTronco: Contiene la foto del tronco del árbol, es un campo de tipo imagen y si se puede dejar en blanco.

fotoFrutos: Contiene la foto de los frutos del árbol, es un campo de tipo imagen y si se puede dejar en blanco.

ubicacion: Contiene la ubicación del árbol, es decir, la zona universitaria donde está ubicado, es un campo de tipo texto, no se puede dejar en blanco y con un tamaño máximo de 50 caracteres.

altura: Contiene la altura del árbol, es un campo de tipo decimal, si se puede dejar en blanco y con un tamaño máximo de 19 caracteres.

perimetro: Contiene el perímetro del árbol, es un campo de tipo decimal, si se puede dejar en blanco y con un tamaño máximo de 19 caracteres.

usuario		CREATE TABLE "principal_usuario" ("idUserio" integer NOT NULL PRIMARY KEY AUTOINCREMENT, "usuario_id" integer NOT NULL UNIQUE REFERENCES "auth_user" ("id") DEFERRABLE INITIALLY DEFERRED)
idUserio	integer	"idUserio" integer NOT NULL PRIMARY KEY AUTOINCREMENT
usuario	integer	"usuario_id" integer NOT NULL UNIQUE

Figura C.5: Tabla Usuario

- **USUARIO:** En este modelo se va a guardar la información relacionada con los usuarios, está compuesta por dos campos que son:

idUserio: Es la clave primaria de la tabla y contiene un identificador único para cada usuario.

usuario: Contiene el nombre del usuario, es una FK.

- **USER:** Este modelo está diseñado por el framework y guarda información relacionada con los usuarios, está compuesta por doce campos que son:

username: Contiene el nombre del usuario, es un campo de tipo texto, tiene que ser único, no se puede dejar en blanco y con un tamaño máximo de 150 caracteres.

first name: Contiene el nombre del usuario, es un campo de tipo texto, se puede dejar en blanco y con un tamaño máximo de 30 caracteres.

last name: Contiene el apellido del usuario, es un campo de tipo texto, se puede dejar en blanco y con un tamaño máximo de 150 caracteres.

email: Contiene el email del usuario, es un campo de tipo email y se puede dejar en blanco.

password: Contiene la contraseña del usuario, es un campo de tipo texto, es necesario y se encripta.

groups: Contiene el grupo al cual pertenece el usuario.

user permissions: Contiene los permisos que tiene cada usuario.

is staff: Nos indica si este usuario puede acceder al sitio de administración.

is active: Nos indica si este usuario está activo.

is superuser: Nos indica si este usuario es super usuario.

last login: Nos indica una fecha y hora del último inicio de sesión del usuario.

date joined: Nos indica una fecha y hora de cuando se creó la cuenta del usuario.

C.3. Diseño procedimental

Para comprender y entender las funciones que va a desempeñar nuestra aplicación web he realizado un diagrama de navegabilidad [C.6](#), el cual explicaré en detalle a continuación:

- En primer lugar el usuario accede a la pantalla principal de la página.
- Una vez dentro tiene dos opciones, registrarse e iniciar sesión o navegar sin las ventajas de ser usuario.

- Si el usuario no inicia sesión podrá:
 - Seguir en Twitter a @UbuVerde.
 - Dar Me Gusta y compartir noticias de UbuVerde.
 - Visitar la página de UbuVerde (Acerca De Nosotros).
 - Ver el mapa con todos los árboles y seleccionar uno y ver sus características y fotos.
 - Ver todas las familias.
 - Elegir una o ver el mapa y seleccionar un árbol.
 - Si seleccionas una familia, la ves y eliges un género.
 - Una vez seleccionado el género, ves el género, la familia y eliges la especie.
 - Ves las características de la especie y eliges el individuo.
 - Ves las fotos, el mapa y las características del árbol seleccionado.
 - Si seleccionas un árbol ves las características del árbol, las fotos y el mapa donde esta ubicado.
 - Ver todos los géneros.
 - Elegir uno o ver el mapa y seleccionar un árbol.
 - Si seleccionas un género, ves la familia y el género seleccionado y eliges una especie.
 - Ves las características de la especie y eliges el individuo.
 - Ves las fotos, el mapa y las características del árbol seleccionado.
 - Si seleccionas un árbol ves las características del árbol, las fotos y el mapa donde esta ubicado.
 - Ver todas las especies.
 - Elegir una o ver el mapa y seleccionar un árbol.
 - Si seleccionas una especie, ves las características de la especie y eliges el individuo.
 - Ves las fotos, el mapa y las características del árbol seleccionado.
 - Si seleccionas un árbol ves las características del árbol, las fotos y el mapa donde esta ubicado.
 - Ver todos los individuos.
 - Ves las 5 ubicaciones y seleccionas una.
 - Ves el mapa con los árboles de esa ubicación y seleccionas uno.

Ves las fotos, el mapa y las características del árbol seleccionado.

- Si el usuario inicia sesión podrá además de todas esas cosas anteriores:
 Descargar un pdf con las características de los árboles.
 Agregar nuevos árboles.
 Cerrar sesión.

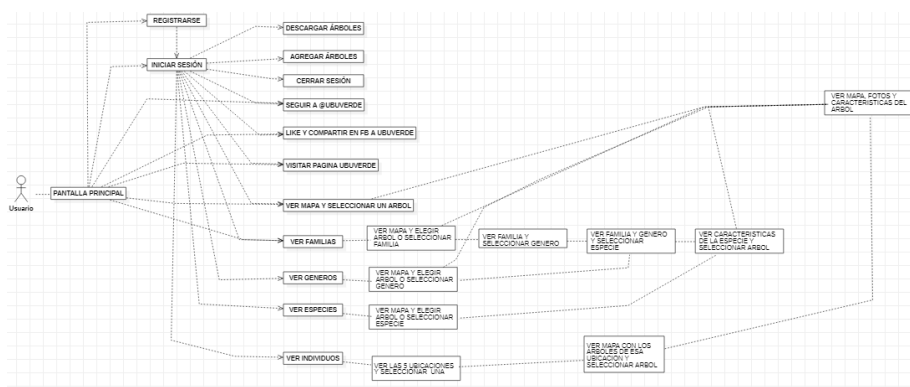


Figura C.6: Diagrama de Navegabilidad

C.4. Diseño arquitectónico

El proyecto como está realizado mediante el framework Django seguirá el patrón MVT[1].

Es un framework muy parecido al MVC, pero con ligeros matices:

- **M**: significa Model o Modelo, es la capa que tiene acceso a la base de datos. En ella esta contenida la información relacionada con los datos, es decir, cómo acceder a ellos, cómo se comportan, cómo se validan y las relaciones entre sí.
- **V**: significa View o Vista, es la capa de la lógica de negocios. En ella esta contenida la lógica que accede al modelo y se comunica con el template o plantilla deseada.
- **T**: significa Template o Plantilla, es la capa de presentación. En ella estará contenido las decisiones que tienen que ver con la presentación de los datos, en nuestro caso como se va a ver la página web.

Muchas veces se tiende a decir que las Vistas de Django se pueden asemejar al controlador y las Plantillas pueden ser las vistas en el MVC, pero esto no es cierto, en realidad la vista en MVC describe los datos que son presentados al usuario; no necesariamente el cómo se mostrarán, pero si cuáles datos son presentados.

Apéndice D

Documentación técnica de programación

- D.1. Introducción
- D.2. Estructura de directorios
- D.3. Manual del programador
- D.4. Compilación, instalación y ejecución del proyecto
- D.5. Pruebas del sistema

Apéndice E

Documentación de usuario

- E.1. Introducción
- E.2. Requisitos de usuarios
- E.3. Instalación
- E.4. Manual del usuario

Bibliografía

- [1] Uniwebsidad. El patrón de diseño mtv. <https://uniwebsidad.com/libros/django-1-0/capitulo-5/el-patron-de-diseno-mtv>, 2020. [Online; accessed 07-Ene-2020].