@startuml

skinparam shadowing false

skinparam componentStyle rectangle

' ===== NODOS CLIENTE =====

node "PC del Usuario" as pcUser <<device>> {

node "Navegador" as browser <<executionEnvironment>>

artifact "Artefacto Web\n(HTML/JS/CSS)" as webapp

browser -down-> webapp : Renderiza

}

node "PC Operador/Admin" as pcAdmin <<device>> {

node "Runtime Desktop\n(.NET/Python/Electron)" as rtDesk <<executionEnvironment>>

artifact "App de Escritorio\n(Admin/Backoffice)" as desktopApp

rtDesk -down-> desktopApp : Ejecuta

}

' ===== BACKEND / SERVIDOR =====

node "App Server" as app <<node>> {

node "Nginx / Reverse Proxy" as nginx <<executionEnvironment>>

node "ASGI (Gunicorn/Uvicorn)" as asgi <<executionEnvironment>>

component "Django\n(API REST / Admin / Views)" as django

nginx -down-> asgi

asgi -down-> django

}

' ===== SERVICIOS GESTIONADOS =====

node "Supabase (Cloud)" as supa <<cloud>> {

database "Postgres" as pg

component "Auth" as auth

component "Storage" as storage

}

node "APIs Externas" as ext <<external>>

' ===== CONEXIONES =====

' Usuario final (web)

pcUser --> nginx : HTTPS 443\n(Navegador consume Web/REST)

' Operador (desktop)

pcAdmin --> nginx : HTTPS 443\n(REST/GraphQL/WebSocket)

' Backend -> servicios

django --> supa : HTTPS+JSON\n(PostgREST/Auth/Storage)

django --> ext : HTTPS 443

' ===== NOTAS =====

note right of desktopApp

App de escritorio:

• Operaciones administrativas

• Catálogo/stock/usuarios

• Tareas batch y monitoreo

• (Opcional) WebSocket para eventos

end note

note right of django

Django expone:

• Admin Panel

• REST/GraphQL

• Webhooks/WS (si aplica)

end note

@enduml