



# **Structura Aplicației & Autentificarea Utilizatorilor**

Versiunea 1.0

## Pagina Principală – Structura Aplicației și Autentificarea Utilizatorilor

1. Descriere generală
2. Structura de fișiere și module principale
3. Logica fișierului principal – app.py
  - a. Inițializarea aplicației
  - b. Mecanismul de autentificare
4. Autentificarea utilizatorilor
  - a. Interfața de login
  - b. Validarea credențialelor
  - c. Protecția interfeței
5. Navigarea în aplicație
6. Funcțiile de încărcare a datelor – data\_helpers.py

Exemple de funcții principale:
7. Concluzii

## Pagina KPI Dashboard – Sumar Executiv

1. Scopul paginii
2. Structura logică a modulului
3. Sursa de date și parametri principali
4. Funcționalități și calcule financiare
  - a. Creanțe (Facturi neîncasate)
  - b. Datorii (Facturi neachitate)
  - c. Stocuri
5. Mecanismul interactiv al sumarului executiv
6. Interfața vizuală

Graficul Waterfall
7. Rolul paginii în arhitectura aplicației
8. Concluzii

## Pagina Vânzări – Brenado For House

1. Scopul paginii
2. Structura logică a modulului
3. Surse de date și prelucrări
4. Funcționalități cheie
  - 4.1. Tab 1 – Vânzări Info Generale
  - 4.2. Tab 2 – Date Detaliat
  - 4.3. Tab 3 – Analize Vizuale
5. Elemente UI & UX
6. Validări și tratament erori
7. Rolul paginii în arhitectură
8. Concluzii

## Pagina Balanță Stocuri – Brenado For House

1. Scopul paginii

- [2. Structura logică a modulului](#)
- [3. Surse de date și prelucrări](#)
- [4. Funcționalități cheie](#)
  - [4.1. Tab 1 – „În Dată”](#)
  - [4.2. Tab 2 – „Analize Stocuri” \(Vechime\)](#)
- [5. Elemente UI & UX](#)
- [6. Validări și tratament erori](#)
- [7. Rolul paginii în arhitectură](#)
- [8. Concluzii](#)

#### [Pagina Cumpărări Intrări – Brenado For House](#)

- [1. Scopul paginii](#)
- [2. Structura logică a modulului](#)
- [3. Surse de date și câmpuri utilizate](#)
- [4. Funcționalități cheie](#)
  - [4.1. KPI Executiv](#)
  - [4.2. Filtrare](#)
  - [4.3. Tabel de lucru \(curățat\)](#)
  - [4.4. Statistici pentru setul filtrat](#)
  - [4.5. Top 20 Furnizori](#)
- [5. Elemente UI & UX](#)
- [6. Validări și tratament erori](#)
- [7. Rolul paginii în arhitectură](#)
- [8. Concluzii](#)

#### [Pagina Facturi Neîncasate – Brenado For House](#)

- [1. Scopul paginii](#)
- [2. Structura logică a modulului](#)
- [3. Surse de date și câmpuri utilizate](#)
- [4. Funcționalități cheie](#)
  - [4.1. KPI-uri executive](#)
  - [4.2. Filtre operative](#)
  - [4.3. Tabel sinteză clienți + selector pentru raport](#)
  - [4.4. Rapoarte PDF](#)
- [5. Logica de calcul & status scadență](#)
- [6. Elemente UI & UX](#)
- [7. Validări „edge cases”](#)
- [8. Rolul paginii în arhitectură](#)
- [9. Concluzii](#)

#### [Pagina Facturi Neachitate – Brenado For House](#)

- [1. Scopul paginii](#)
- [2. Structura logică a modulului](#)
- [3. Surse de date și câmpuri utilizate](#)

- 4. Funcționalități cheie
  - 4.1. Comutator „Cu Efecte”
  - 4.2. KPI
  - 4.3. Filtre operative
  - 4.4. Sinteză pe Furnizor + selector de includere
- 5. Raportare PDF (continuu, multi-furnizor)
  - 5.1. Header unic
  - 5.2. Bloc furnizor (în corp, fără PageBreak)
  - 5.3. Total pe întregul raport
- 6. Logica de scadență și status
- 7. Elemente UI & UX
- 8. Validări & robustețe
- 9. Concluzii

Pagina Scadențe Plăți cu Efecte – Brenado For House

- 1. Scopul paginii
- 2. Localizare și protecție acces
- 3. Surse de date și câmpuri necesare
- 4. Indicatori principali (KPI)
- 5. Filtrare date
  - 5.1. Data scadenței (multiselect)
  - 5.2. Terț (multiselect)
- 6. Tabel de lucru
- 7. Vizualizări grafice
  - 7.1. Distribuție pe Zile (bar stacked)
  - 7.2. Heatmap Calendar (ziua lunii × luna)
  - 7.3. Distribuție Terți (donut + tabel)
  - 7.4. Timeline (bubble + cumulativ)
- 8. Comportamente și validări
- 9. Beneficii operaționale
- 10. Concluzie

# Pagina Principală – Structura Aplicației și Autentificarea Utilizatorilor

## 1. Descriere generală

Aplicația **Brenado For House / Brenado**, dezvoltată de **CASTEMILL SRL**, este o platformă analitică web destinată companiilor care doresc centralizarea și vizualizarea indicatorilor financiari și operaționali.

Soluția este construită pe tehnologia **Streamlit**, utilizând limbajul **Python** și pachetele **pandas**, **plotly**, și **openpyxl** pentru procesarea datelor din fișiere Excel exportate zilnic din ERP-ul companiei.

Structura aplicației este modulară și permite gestionarea a două entități economice distincte („Brenado For House” și „Brenado”), fiecare având propriile sale module de analiză.

---

## 2. Structura de fișiere și module principale

Aplicația este organizată în mai multe componente funcționale:

- **app.py** – fișierul principal care gestionează interfața de navigare, autentificarea și logica de încărcare a paginilor.
  - **utils/auth\_helper.py** – modul care gestionează procesul de autentificare a utilizatorilor.
  - **utils/data\_helpers.py** – modul de prelucrare și încărcare a datelor din fișierele Excel (preluate din ERP).
  - **pages/** – directorul care conține toate paginile aplicației (KPI, Vânzări, Stocuri, Facturi etc.).
  - **data/** – folderul unde sunt stocate fișierele Excel zilnice.
  - **api/** – componenta de interfațare cu Firebase și webhookurile pentru managementul pontajului și al echipelor de lucru.
- 

## 3. Logica fișierului principal – app.py

Fișierul principal **app.py** controlează încărcarea aplicației, autentificarea și navigarea între module.

### a. Inițializarea aplicației

La pornire, aplicația verifică existența logo-ului companiei în calea data/logo/logobre2.png.

Dacă imaginea este găsită, aceasta este folosită ca pictogramă a aplicației (favicon).

În caz contrar, se utilizează un fallback standard cu emoji 🏠.

Totul este gestionat prin `st.set_page_config()` pentru stabilirea titlului, iconiței și layout-ului („wide”).

## b. Mecanismul de autentificare

După configurarea paginii, aplicația setează o variabilă de sesiune:

```
python

st.session_state['page_key'] = 'main'
verificare_autentificare()
```

Funcția `verificare_autentificare()` este apelată din modulul `auth_helper.py` și are rolul de a restricționa accesul la conținutul aplicației până la introducerea corectă a credențialelor.

---

# 4. Autentificarea utilizatorilor

Procesul de autentificare este centralizat în fișierul `utils/auth_helper.py`.

La accesarea oricărei pagini, aplicația verifică dacă sesiunea conține un parametru autentificat = True.

Dacă nu, utilizatorul este direcționat automat către formularul de autentificare.

## a. Interfața de login

Formularul este afișat centralizat, conținând:

- câmp „Utilizator”
- câmp „Parolă”
- buton „Login”

## b. Validarea credențialelor

Datele sunt comparate cu valorile stocate în fișierul `.streamlit/secrets.toml`:

python

```
user_corect = st.secrets["credentials"]["username"]  
pass_corecta = st.secrets["credentials"]["password"]
```

Dacă autentificarea reușește, sesiunea este marcată ca activă și aplicația se reîncarcă automat prin `st.rerun()`.

În caz contrar, apare mesajul „Utilizator sau parolă greșită”.

### c. Protecția interfeței

CSS-ul încorporat ascunde complet elementele Streamlit standard (meniul, bara de titlu și butonul de deploy), oferind un aspect curat și profesional:

CSS

```
.stApp > header {visibility: hidden;}  
.stDeployButton {display: none;}  
#MainMenu {visibility: hidden;}
```

---

## 5. Navigarea în aplicație

După autentificare, utilizatorul accesează interfața principală cu două secțiuni distincte:

### 1. Brenado For House

- KPI Dashboard (`pages/KPI.py`)
- Vânzări (`pages/vanzari.py`)
- Balanță Stocuri (`pages/balanta_stocuri.py`)
- Cumpărări Intrări (`pages/cumparari_intrari.py`)
- Facturi Neîncasate (`pages/facturi_neincasate.py`)
- Facturi Neachitate (`pages/facturi_neachitate.py`)
- Scadențe Plăți (`pages/scadente_plati.py`)

### 2. Brenado

- Vânzări (`pages/vanzariB.py`)
- Balanță Stocuri (`pages/balanta_stocuri_B.py`)
- Facturi Neîncasate / Neachitate (`pages/facturi_neincasate_B.py`,  
`pages/facturi_neachitate_B.py`)
- Scadențe Plăți (`pages/scadente_plati_B.py`)

Navigarea este controlată prin componenta `st.navigation(pages)` din Streamlit, care permite rularea paginii selectate din dicționarul `pages`.

---

## 6. Funcțiile de încărcare a datelor – data\_helpers.py

Fișierul `utils/data_helpers.py` conține funcțiile de prelucrare a fișierelor Excel exportate zilnic din ERP-ul clientului.

Fiecare funcție este decorată cu `@st.cache_data`, pentru a reduce timpul de încărcare și a evita citirea repetată a acelorași fișiere.

### Exemple de funcții principale:

- `load_vanzari()` – încarcă fișierele de vânzări lunare din `data/vanzari/YYYY/MM_luna.xlsx`.
- `load_balanta_la_data()` – încarcă balanța de stocuri actuală.
- `load_neincasate()` / `load_neachitate()` – extrag datele financiare privind creanțele și datoriile.
- `load_cumparari_intrari()` – listează intrările de marfă și achizițiile recente.
- `load_scadente_plati()` – gestionează plățile și efectele comerciale.
- Versiunile marcate cu sufixul **B** (ex: `load_vanzariB()`, `load_neincasateB()`) sunt dedicate companiei **Brenado**.

În caz de eroare sau fișiere lipsă, fiecare funcție returnează un **DataFrame demo** pentru a permite testarea aplicației fără date reale.

---

## 7. Concluzii

Pagina principală a aplicației **Brenado For House** reprezintă scheletul întregului sistem.

Aici sunt definite:

- structura aplicației și navigarea între module,
- mecanismul de autentificare securizată,
- funcțiile de încărcare a datelor (ETL din ERP),
- logica de inițializare a sesiunii.

Acest design modular permite extinderea facilă a aplicației pentru alte companii, cu fișiere și date proprii, fără a modifica structura de bază.

Aplicația a fost proiectată pentru performanță, claritate vizuală și ușurință în utilizare, oferind un tablou de bord complet pentru analiza activității economice zilnice.



# Pagina KPI Dashboard – Sumar Executiv

## 1. Scopul paginii

Pagina **KPI Dashboard** reprezintă modulul central al aplicației **Brenado For House**, oferind o imagine sintetică asupra principalilor indicatori financiari: creanțe, datorii, stocuri și lichiditate.

Aceasta este prima interfață accesibilă utilizatorului după autentificare și servește drept „**Sumar Executiv**” pentru situația economică a companiei.

Obiectivul principal al paginii este de a corela automat datele provenite din mai multe surse (facturi neîncasate, facturi neachitate, balanța de stocuri) și de a calcula indicatorii de performanță cheie în timp real, pe baza fișierelor zilnice încărcate din ERP.

---

## 2. Structura logică a modului

Fișierul **pages/KPI.py** este construit modular și include următoarele componente principale:

### 1. Inițializare și configurare pagină

- Setarea layoutului Streamlit la mod „wide”.
- Afișarea titlului: Dashboard KPI - Brenado For House.
- Încărcarea funcției de autentificare din `auth_helper.py` pentru protecția accesului.

### 2. Încărcarea datelor

- Folosește funcțiile din `utils.data_loaders`:
  - `load_neincasate()` – încarcă facturile neîncasate (creanțe).
  - `load_neachitate()` și `load_neachitate_total()` – încarcă facturile neachitate, diferențiate între cele „cu efecte” și „fără efecte”.
  - `load_balanta_la_data()` – preia valoarea actuală a stocurilor.

### 3. Calcularea indicatorilor

- Pentru fiecare sursă de date, există funcții dedicate:
  - `calculate_neincasate_metrics()`
  - `calculate_neachitate_metrics()`
  - `calculate_stocuri_metrics()`
- Fiecare funcție returnează valori sintetice precum totaluri, solduri sau sume achitate.

### 4. Prezentarea sumarului executiv

- Datele sunt agregate într-un tablou sintetic afișat pe trei coloane:
  - Active circulante (creanțe + stocuri)
  - Datorii curente
  - Indicatori cheie de lichiditate

#### 5. Vizualizare grafică

- Este generat un grafic tip **Waterfall (Plotly)**, care vizualizează fluxul financiar între active, datorii și lichiditatea netă.

### 3. Sursa de date și parametri principali

Pagina se bazează pe datele prelucrate anterior din ERP și salvate sub formă de fișiere Excel:

Tip date	Fișier sursă	Funcție de încărcare
Facturi neîncasate	data/FacturiNeincasate.xlsx	load_neincasate()
Facturi neachitate (cu efecte)	data/FacturiNeachitate.xlsx	load_neachitate()
Facturi neachitate (fără efecte)	data/FacturiNeachitateTotal.xlsx	load_neachitate_total()
Balanță stocuri	data/LaData.xlsx	load_balanta_la_data()

Valoarea TVA utilizată în calcule este definită global.

Aceasta se aplică asupra stocurilor pentru a reflecta valoarea comercială brută.

### 4. Funcționalități și calcule financiare

#### a. Creanțe (Facturi neîncasate)

Funcția `calculate_neincasate_metrics()`:

- preia fișierul de facturi neîncasate,
- convertește datele în format `datetime` (`Data`, `DataScadenta`),

- calculează totalul, soldul și sumele deja achitate.

Rezultatul este o structură de tip:

```
python  
  
{ 'total': 50000, 'sold': 32000, 'achitat': 18000 }
```

## b. Datorii (Facturi neachitate)

Funcția `calculate_neachitate_metrics()` calculează separat datoriile „cu efecte” și „fără efecte”.

Astfel se pot analiza situațiile financiare fie cu instrumente comerciale (bilete la ordin, cambii), fie doar datoriile curente.

## c. Stocuri

Funcția `calculate_stocuri_metrics()` preia balanța curentă și extrage valorile „Stoc final” și „Valoare de vânzare”.

În sumarul executiv, stocurile sunt ajustate automat cu TVA pentru o evaluare completă a activelor.

---

# 5. Mecanismul interactiv al sumarului executiv

Utilizatorul poate alege:

- **Tipul datoriilor** folosit la calculul lichidității:
  - *Cu efecte* sau *Fără efecte*.
- **Baza de evaluare a stocurilor:**
  - *Valoare stoc (cost)* sau *Valoare de vânzare*.

Aceste alegeri actualizează automat valorile afișate în sumar.

Indicatorii calculați sunt:

- **Active circulante** = Creanțe + Stocuri (cu TVA)
- **Lichiditate netă** = Active circulante – Datorii curente
- **Rata lichidității** = Active circulante / Datorii curente

---

# 6. Interfața vizuală

Tabloul principal este împărțit în trei coloane:

1. **Active Circulante** – prezintă totalul creanțelor și al stocurilor.
2. **Datorii Curente** – diferențiate între „cu efecte” și „fără efecte”.
3. **Indicatori Cheie** – afișează lichiditatea netă și statusul financiar („POZITIV” sau „ATENȚIE”).

## Graficul Waterfall

Graficul generat cu biblioteca **Plotly** oferă o reprezentare vizuală a fluxului financiar:

- Bara 1: Active Circulante
- Bara 2: Încasări Realizate
- Bara 3: Datorii Curente
- Bara 4: Lichiditate Netă

Această vizualizare ajută la înțelegerea rapidă a echilibrului între resurse și obligații, fiind esențială pentru luarea deciziilor de management.

---

## 7. Rolul paginii în arhitectura aplicației

Pagina KPI este centrul de control al aplicației.

Ea integrează datele din toate modulele, oferind un tablou de bord unitar pentru analiza financiară.

Indicatorii calculați sunt folosiți ca referință pentru restul paginilor (vânzări, facturi, stocuri).

Modulul este conceput pentru a fi **scalabil**, putând prelua și date istorice sau fișiere suplimentare fără modificări semnificative în cod.

---

## 8. Concluzii

Modulul **Dashboard KPI – Sumar Executiv** oferă o imagine sintetică, clară și complet automatizată a performanței financiare zilnice.

Prin corelarea datelor din ERP și prezentarea interactivă a indicatorilor, aplicația permite managementului companiei **Brenado For House** să vizualizeze în timp real starea lichidității, gradul de acoperire a datoriilor și evoluția activelor circulante.

Această pagină definește fundamentul analitic al întregii aplicații și asigură un **punct unic de control financiar**, fiind primul nivel de analiză înaintea modulelor detaliate (vânzări, stocuri, facturi etc.).

# Pagina Vânzări – Brenado For House

## 1. Scopul paginii

Pagina **Vânzări** oferă managementului o vedere operațională și analitică asupra performanței comerciale: evoluția vânzărilor pe zile/luni, top clienți și produse, analiză pe gestiuni și agenți, plus export rapid al datelor filtrate în Excel.

Modulul face legătura dintre datele încărcate zilnic din ERP (fișiere Excel) și deciziile de business, prin filtre dinamice, metrici cheie și vizualizări interactive.

---

## 2. Structura logică a modulului

Fișier: `pages/vanzari.py`

1. **Protecție acces & context timp**
    - `st.session_state['page_key'] = 'vanzari'` și `verificare_autentificare()` asigură accesul doar pentru utilizatori autentificați.
    - Setare timezone: Europe/Bucharest și definirea unui **prag TVA**: `VAT_CHANGE_DATE = 2025-08-01`.
  2. **Încărcare date**
    - `load_vanzari_istorice()` – încarcă toate fișierele istorice din `data/vanzari/YYYY/MM_luna.xlsx`.
    - `load_vanzari()` – încarcă fișierul curent (luna prezentă).
    - Conversie sigură a coloanei Data la datetime prin `ensure_datetime_column()`.
  3. **Tab-uri principale**
    - **Tab 1: Vânzări Info Generale** – metrici rapide (azi, luna curentă, anul curent, perioadă selectată) și comutator valori **net/brut** (fără/ cu TVA).
    - **Tab 2: Date Detaliate** – filtre, tabel interactiv, top 20 clienți (listă/treemap) și **export Excel**.
    - **Tab 3: Analize Vizuale** – comparații lunare (an curent vs an anterior), totaluri și **Top 10 clienți** pe interval.
- 

## 3. Surse de date și prelucrări

- **Sursă principală:** fișiere Excel preluate din ERP, în structura:
  - `data/vanzari/YYYY/MM_luna.xlsx` (istoric)
  - `data/vanzari/2025/08_august.xlsx` (exemplu pentru luna curentă)
- **Conversii:**
  - Data → datetime (tolerant la erori).
  - Numerice importante: Valoare, Cost, Adaos, Cantitate, Pret → numeric cu fallback la 0.

- **TVA variabil:**  
Funcția `calculate_net_sum()` aplică TVA per linie, în funcție de dată:
    - 19% înainte de **01.08.2025**
    - 21% după și inclusiv **01.08.2025**
 Comutatorul de la utilizator permite afișare **netă** (fără TVA) sau **brută** (cu TVA).
- 

## 4. Funcționalități cheie

### 4.1. Tab 1 – Vânzări Info Generale

- **Comutator valori:** „Activează pt valori brute (cu TVA)” – calcule pe net/brut.
- **Metrici:**
  - **Azi** – total, cost și adaos pe data curentă.
  - **Luna curentă** – total, cost, adaos agregate pe luna selectată din calendar.
  - **Anul curent** – total, cost, adaos pe întreg anul.
  - **Perioadă selectată** – selector de interval (de la începutul lunii curente până la azi, implicit).

### 4.2. Tab 2 – Date Detaliat

- **Filtre dinamice:**
  - Gestiune, Client, Agent, Produs (multiselect), Interval de date.
- **Tipuri de afișare:**
  - *Standard*: tabel cu înregistrări sortate descrescător după dată.
  - *Zi și Clienți*: agregare pe (Data, Client) cu sume pentru Valoare și Adaos.
  - *Top Produe*: listă cu Denumire, Cantitate, Valoare, Adaos.
- **Tabel interactiv**: randare pregătită pentru compatibilitate (evitarea erorilor Arrow) prin `fix_dataframe_for_display()`.
- **Metrici sumare**: total **Valoare** și **Adaos** pentru setul filtrat.
- **Export Excel (un click):**
  - Buton „**Descarcă tabelul filtrat (Excel)**” – generează fișier .xlsx cu auto-fit pe coloane.
- **Top 20 Clienți – anul curent:**
  - Alegere între „*Listă compactă*” (cu pondere %) sau „*Treemap*” (Plotly).
  - Sume pe client: Valoare, Adaos.

### 4.3. Tab 3 – Analize Vizuale

- **Analiză vânzări pe perioade:**
  - Selectoare: **An** și **Lună** (din datele existente).
  - **Comparație vs anul trecut** (dacă există date); aliniază graficul prin suprapunere (linia anului anterior este *dash*).
  - Grafic **linie** (Plotly) cu vânzări zilnice agregate.
  - **Total lunar** și, dacă se compară, **delta %** față de aceeași perioadă a anului trecut (ține cont dacă luna curentă e completă sau nu).
- **Top 10 Clienți (bară orizontală):**

- Selectarea intervalului de analiză direkt din UI.
  - Bar chart interactiv (Plotly), cu hover detaliat.
- 

## 5. Elemente UI & UX

- Toate componentele sunt **responsive** și folosesc layout „wide”.
  - Hover-urile grafice includ formatare numerică în **RON**, cu separatori de mii.
  - Legendă orizontală și „hovermode: x unified” pentru comparații clare în graficele linie.
  - Tabelele sunt dimensionate pentru lizibilitate (înălțime fixă ~390 px în Tab 2).
- 

## 6. Validări și tratament erori

- Conversiile la datetime folosesc `errors='coerce'` – valorile invalide devin NaT și sunt excluse din calcule.
  - Dacă lipsesc coloanele necesare pentru un anumit view (ex. Client, Valoare, Adaos, Data), UI afișează **mesaj informativ** și evită oprirea întregii pagini.
  - La lipsa datelor pe intervalele alese, se afișează „*Nu există date pentru intervalul selectat.*”
  - Exportul Excel se bazează pe `xlsxwriter`, cu auto-fit pe coloane pentru lizibilitate.
- 

## 7. Rolul paginii în arhitectură

Modulul **Vânzări** este nucleul analitic pentru zona comercială.

El este folosit pentru:

- verificarea performanței zilnice/lunare/anuale,
- identificarea clienților cheie,
- evaluarea mixului de produse,
- suport rapid pentru rapoarte (prin export Excel).

Prin structura de filtre și vizualizări, pagina permite atât **overview executiv**, cât și **analiză granulară** la nivel de client, produs sau agent.

---

## 8. Concluzii

Pagina **Vânzări** îmbină **ușurința în utilizare** cu **analiza avansată**:

metrici rapide, filtre multiple, grafice comparative și export instant de date.

Datorită **calculului dinamic al TVA** și al **comparațiilor inter-an**, modulul oferă o imagine realistă și actualizată a performanței comerciale, devenind un instrument esențial pentru deciziile de business.

# Pagina Balanță Stocuri – Brenado For House

## 1. Scopul paginii

Pagina **Balanță Stocuri** oferă o vedere completă asupra inventarului: valoarea la cost vs valoarea de vânzare, distribuția stocurilor pe gestiuni, producători și produse, precum și **analiza vechimii stocurilor** pe intervale (1–90 zile, 90–365 zile, >365 zile).

Modulul combină metrici cheie, filtre interdependente și vizualizări (donut, treemap) cu **export rapid în Excel** pentru analize și raportări.

---

## 2. Structura logică a modulului

Fișier: **pages/balanta\_stocuri.py**

### 1. Protecție acces

`st.session_state['page_key'] = 'stocuri' + verificare_autentificare()` asigură accesul exclusiv utilizatorilor autentificați.

### 2. Helperi și reutilizabile

- **Carduri HTML de KPI:** `render_metric_card()`, `render_metrics_row()` pentru afișarea valorilor cheie.
- **Filtrare:** `apply_filters()` (aplică `.isin` pe coloane), `safe_column_check()` (valori unice sigure), `filter_and_display_table()` (tabel cu coloane selectate).
- **Calcul metrici:** `calculate_metrics(df, columns)` — sume simple pentru coloanele numeric importante.
- **Export Excel:** `to_excel_bytes(df, sheet_name)` — generează fișier `.xlsx` din cadrul UI.
- **Vizualizări:** `create_donut_chart()` — pie/donut cu total central.
- **Card compact pe gestiune:** `render_gestiune_compact_card()` — afișează, într-un singur card, **Total** și defalcarea pe **1–90 / 90–365 / >365** pentru fiecare gestiune.

### 3. Tab-uri funcționale

- **Tab 1 – „În Dată”:** situația inventarului la zi (cost vs vânzare), filtre (gestiune → producător → produs), tabel și distribuție pe gestiuni (donut).



- **Tab 2 – „Analize Stocuri”**: **vechimea stocurilor** pe intervale, KPI-uri pe gestiuni (card unic/gestiune), distribuții pe intervale/grupă/gestiune (pie + treemap) și **Explorer produse** (cu export XLSX).
- 

### 3. Surse de date și prelucrări

- **Balanță la dată**: `load_balanta_la_data()` (ex. `data/LaData.xlsx`)  
Coloane utilizate uzual: `DenumireGest`, `Denumire`, `UM`, `Pret`, `Stoc final`, `ValoareStocFinal`, `PretVanzare`, `ValoareVanzare`, `Producator`.
  - **Vechime stocuri**: `load_balanta_vechime()` (ex. `data/VechimeStocuri.xlsx`)  
Coloane esențiale: `Denumire gestiune`, `Grupa`, `Denumire`, `Cod`, `UM`, `Data IN`, `ZileVechime`, `Pret`, `Pret vanzare`, `Stoc final`, `Valoare intrare`.
  - **Mapare vechime** → **intervale** (exact 3 categorii):
    - 1–90 zile
    - 90–365 zile
    - >365 zileStocurile sunt analizate preponderent la **valoarea de intrare (achiziție)** pentru consistență în comparații.
  - **Curățare numerică**: câmpurile valorice sunt convertite la tip numeric cu toleranță la erori.
- 

### 4. Funcționalități cheie

#### 4.1. Tab 1 – „În Dată”

- **KPI-uri imediate** (carduri HTML):
  - **Total Valoare Cost** (`ValoareStocFinal`)
  - **Total Valoare Vânzare** (`ValoareVanzare`)
- **Filtre interdependente**:
  - **Gestiune** → 2) **Producător** (filtrat după gestiune) → 3) **Produs** (filtrat după producător)  
Filtrele se aplică cumulativ prin `apply_filters()`.
- **Tabel de lucru** (coloane cheie):  
`DenumireGest`, `Denumire`, `UM`, `Pret`, `Stoc final`, `PretVanzare`, `Producator`  
— randat cu `st.dataframe()` pentru explorare rapidă.
- **KPI filtrate** (dacă există selecții):
  - **Total Valoare Stoc Final Filtrată**
  - **Total Valoare Vânzare Filtrată**
- **Distribuție pe gestiuni** (donut) — când e ales minim un produs:  
sumă **Stoc final** pe `DenumireGest`; totalul este afișat central.

#### 4.2. Tab 2 – „Analize Stocuri” (Vechime)

- **Normalizare vechime**: `ZileVechime` → Interval vechime (3 categorii ordonate).

- **KPI pe gestiuni** (card unic/gestiune):  
pentru gestiunile-țintă (ex.: *Craiova Toamnei*, *Depozit Grele Galicea*, *Showroom Galicea Mare*):
    - **Total Valoare Stoc (achiziție)**
    - **1–90 zile / 90–365 zile / >365 zile**
  - **Distribuție pe intervale pentru o grupă dintr-o gestiune:**  
selectoare **Gestiune + Grupă** ⇒ **donut (pie cu „hole”)** cu total central și culori consistente:
    - verde: **1–90 zile**
    - galben: **90–365 zile**
    - roșu: **>365 zile**
  - **Treemap ierarhic (Interval → Gestiune → Grupă):**  
accent pe „**Valoare intrare**”, culori consistente pe intervale; nivel maxim 3 (fără produse), pentru lizibilitate.
  - **Explorer produse pe intervale** (tabs: *Toate*, *1–90*, *90–365*, *>365*):
    - sortare: Valoare intrare desc, apoi ZileVechime asc;
    - coloane utile (din cele disponibile): Denumire gestiune, Grupa, Denumire, Cod, UM, ZileVechime, Stoc final, Pret, Pret vanzare, Valoare intrare, Interval vechime;
    - **Export XLSX**: buton dedicat per tab, cu denumire de fișier care include intervalul și timestamp-ul.
- 

## 5. Elemente UI & UX

- **Carduri KPI** cu culori coerente (cost vs vânzare; vechime verde/galben/roșu).
  - **Filtre progresive** pentru a ghida utilizatorul și a evita selecții invalide.
  - **Donut și treemap** cu totaluri vizibile, legendă explicită și hover clar (valorile în RON sau unități).
  - **Tabele compacte** cu coloane relevante și export dintr-un click.
  - Layout **responsive**, potrivit pentru ecrane înguste (wrap pe cardul compact de gestiune).
- 

## 6. Validări și tratament erori

- La lipsa datelor (ex. fișier inexistent), UI afișează mesaje **informative** fără a bloca întreaga pagină.
  - Conversiile numerice/date folosesc `errors="coerce"` — valorile invalide sunt ignorate în calcule.
  - Verificări de existență a coloanelor înainte de agregări; dacă lipsesc, se notifică utilizatorul.
  - Exportul XLSX gestionează **auto-fit** simplificat al coloanelor și nume de fișiere safe (fără caractere speciale).
- 

## 7. Rolul paginii în arhitectură

Modulul **Balanță Stocuri** completează tabloul financiar oferit de KPI și Vânzări, răspunzând la întrebări precum:

- Care este **valoarea totală a stocurilor** la cost și la preț de vânzare?
  - Cum se **distribuie stocurile** între gestiuni/producători/produse?
  - Ce **proporție a stocului** este imobilizată în intervale mari de vechime (>365 zile)?
  - Unde sunt **oportunitățile de lichidare** sau **riscurile de depreciere**?
- 

## 8. Concluzii

Pagina **Balanță Stocuri** oferă un instrument clar și complet pentru **managementul inventarului**:

KPI-uri executive, filtre inteligente, vizualizări puternice și **export instant** în Excel.

Prin analiza vechimii pe intervale standardizate și vizualizări ierarhice (treemap), modulul susține decizii rapide privind **rotația stocurilor**, **aprovizionarea** și **promovarea** produselor lente.

# Pagina Cumpărări Intrări – Brenado For House

## 1. Scopul paginii

Modulul **Cumpărări Intrări** oferă o vizibilitate clară asupra aprovizionărilor: ce produse au intrat în gestiuni, în ce cantități, la ce prețuri și de la ce furnizori.

Pagina combină **filtre rapide**, **metrice executive**, un **tabel curat** pentru lucru de zi cu zi și o secțiune de **Top 20 Furnizori** (listă sau treemap), plus **export Excel**.

---

## 2. Structura logică a modulului

Fișier: `pages/cumparari_intrari.py`

### 1. Protecție acces

`st.session_state['page_key'] = 'cumparari'` și `verificare_autentificare()` blochează accesul până la autentificare.

2. **Helper UI**
    - `render_metric_card(label, value, color, suffix, decimals)` – card HTML pentru KPI-uri (formatări cu separatori de mii, sufix „RON” etc.).
  3. **Încărcare date**
    - `cumparari_df = load_cumparari_intrari()` – citește fișierul Excel cu intrările în stoc.
    - Conversie dată: dacă există coloana Data, este transformată la datetime (`dayfirst=True`) și se creează `Data_Display` (format DD/MM/YYYY).
  4. **Calcul KPI**
    - Total 2025 (suma din coloana Valoare, dacă există).
  5. **Filtre**
    - **Gestiune** (multiselect)
    - **Furnizor** (multiselect)
    - **Interval de date** (single-date sau start–end, în funcție de input)
  6. **Tabel de lucru**
    - Coloane selectate și redenumite pentru claritate: Gestiune, Denumire, UM, Cantitate, Preț, Data, Furnizor
    - Formatări pentru Cantitate și Preț.
  7. **Statistici pe setul filtrat**
    - **Total Filtrat** (suma Valoare pentru rândurile vizibile) + **număr intrări** (intern).
  8. **Top 20 Furnizori**
    - Agregare pe Furnizor → Total (RON) + Pondere (% din totalul topului).
    - Vizualizare: **Listă compactă** sau **Treemap** (Plotly).
- 

### 3. Surse de date și câmpuri utilizate

- `load_cumparari_intrari()` (ex.: `data/CumparariIntrariInStoc.xlsx`)
- Câmpuri tipice folosite:  
Gestiune, Denumire, UM, Cantitate, Pret, Valoare, Data, Furnizor
- Câmpuri derivate pentru UI:  
`Data_Display` (format text pentru tabel)

Notă: dacă una dintre coloanele cheie lipsește, pagina afișează un mesaj cu **coloanele disponibile** și prezintă totuși datele brute pentru diagnoză.

---

## 4. Funcționalități cheie

### 4.1. KPI Executiv

- **Total 2025** – suma Valoare pe datasetul încărcat.  
Randat ca **card HTML** cu accent vizual .

### 4.2. Filtrare

- **Gestiune** – selectezi una sau mai multe gestiuni.
- **Furnizor** – selectezi unul sau mai mulți furnizori.
- **Data** – interval (start–end) sau o singură zi; implicit se propune **cea mai relevantă dată din interval** (azi dacă e în interval, altfel capătul cel mai apropiat).

### 4.3. Tabel de lucru (curățat)

- Coloane afișate: Gestiune, Denumire, UM, Cantitate, Preț, Data, Furnizor.
- Formatări:
  - Cantitate și Preț sunt afișate cu două zecimale și separatori de mii.
  - Data este în format DD/MM/YYYY.
- Tabelul este **responsive** și optimizat pentru lucru zilnic.

### 4.4. Statistici pentru setul filtrat

- **Total Filtrat** – suma Valoare pentru rândurile rămase după filtrare (card HTML).
- (Intern) număr intrări – pentru verificări rapide.

### 4.5. Top 20 Furnizori

- Agregare: sum(Valoare) pe Furnizor → Total
- Se calculează **Ponderea (%)** în totalul topului.
- Vizualizare:
  - **Listă compactă** (tabel cu Loc, Furnizor, Total, Pondere)
  - **Treemap** (Plotly) cu hover „Total: ... RON”, scară cromatică albastră, legendă implicită.

## 5. Elemente UI & UX

- **Carduri KPI** curate, cu palate cromatice coerente.
- **Mesaje prietenoase** când lipsesc date sau coloane, pentru diagnoză.
- **Tabel clar** pentru copy/paste sau screenshot în rapoarte operative.
- **Treemap** pentru comunicare vizuală a concentrației pe furnizori.
- Dimensiuni grafice optimizate pentru lizibilitate (≈540px înălțime treemap).

## 6. Validări și tratament erori

- Dacă `cumparari_df` este gol, UI afișează **avertisment** și oprește randarea secțiunilor dependente.
- Pentru coloanele lipsă, pagina listează **coloanele disponibile** (pentru mapare/ETL) și continuă cu o afișare generală, evitând crash-ul.
- Conversiile numerice folosesc `errors='coerce'` și valori implicite sigure.

## 7. Rolul paginii în arhitectură

Modulul **Cumpărări Intrări** completează analiza comercială alături de **Vânzări** și **Balanță Stocuri**, oferind răspunsuri la întrebări cheie:

- Care este **volumul total** al intrărilor pe perioade/gestiuni/furnizori?
- Ce **furnizori concentrează** cea mai mare valoare?
- Cum evoluează **prețurile și cantitățile** la nivel de produs?

Pagina este utilă pentru **aprovizionare**, **negocierea cu furnizorii** și **control bugetar**.

---

## 8. Concluzii

**Cumpărări Intrări** oferă un cadru complet pentru monitorizarea aprovizionării:

**KPI executiv**, **filtre precise**, **tabel curat** pentru operațional și **analize de furnizori** cu treemap.

Prin tratarea robustă a erorilor și UI-ul simplu, modulul este pregătit pentru lucru zilnic și integrare în rapoarte manageriale.

# Pagina Facturi Neîncasate – Brenado For House

## 1. Scopul paginii

Modulul **Facturi Neîncasate** centralizează creanțele comerciale pe client, oferind:

- situația completă a facturilor (în termen / scadente azi / restante),
- totaluri și filtre operative,
- **rapoarte PDF** individuale și multi-client, gata de trimis către clienți sau agenți.

Această versiune corectată afișează **numărul real al facturii** (din coloana *Serie*) și păstrează toate coloanele necesare în filtrări/rapoarte.

---

## 2. Structura logică a modulului

Fișier: `pages/facturi_neincasate.py`

1. **Protecție acces**  
`st.session_state['page_key'] = 'neincasate' + verificare_autentificare()`.
  2. **Setări & constante**
    - Fus orar: **Europe/Bucharest** (pentru date/ore corecte în rapoarte).
    - **LISTĂ EXCLUDERI CLIEŢI** (`CLIENTI_EXCLUSI`) – opțional, pentru a elimina clienți speciali din calcule/afișare (bifă în UI).
  3. **Încărcare & pregătire date**
    - `load_neincasate()` → conversie `Data` și `DataScadenta` la `datetime`.
    - **display\_df\_full**: păstrează toate coloanele; din el se derivă **display\_df\_base** (după excluderi, dacă sunt bifate).
  4. **Helperi**
    - `render_metric_card(label, value, color)` – card KPI.
    - `get_invoice_number(row)` – returnează **numărul real** al facturii din coloana `Serie`.
  5. **Generator PDF**
    - **Individual**: `generate_pdf_report(client_data, client_name)`
    - **Multi-client**: `generate_multi_client_pdf_report(clients_data_dict, agent_name=None)`
- 

### 3. Surse de date și câmpuri utilizate

- Sursă: `load_neincasate()` (ex.: `data/FacturiNeincasate.xlsx`).
  - Câmpuri uzuale: `Client`, `Agent`, `Data`, `DataScadenta`, `Serie` (număr factură), `Total`, `Sold`, `Valuta`, `PL` etc.
  - **Notă**: *Numărul facturii* se citește din **Serie**, nu se mai re-generează, decât dacă lipsește în date.
- 

## 4. Funcționalități cheie

### 4.1. KPI-uri executive

- **Sold** (după excluderi, dacă bifa este activă).
- **Sold Clienți Listă** (valoarea cumulată a clienților excluși; afișat când excluderea este activă).

### 4.2. Filtre operative

- **Client** (multiselect)
- **Agent** (multiselect)
- **Scadențe până la data curentă** – bifa „Data scadență până la DD/MM/YYYY” (arată doar facturile scadente/depășite până azi).
- **Excluderi** – bifa „Elimină clienții din lista specială” (recalculează totul pe baza setului rămas).

### 4.3. Tabel sinteză clienți + selector pentru raport

- Agregare pe **Client** → Sold total.
- **Editor interactiv** (coloană *Selectat*) pentru a marca clienții care intră în raportul multi-client.

## 4.4. Rapoarte PDF

1. **Previzualizare PDF (selecție)**
  - Construiește **Raport Multi-Client** doar pentru clienții bifați.
  - Include: antet (logo, titlu, data), bloc date furnizor + client, tabel facturi, **total restante** și **total general**.
  - Afișează status: *RESTANTĂ (x zile)*, *SCADENTĂ AZI*, *ÎN TERMEN (x zile)*.
2. **Raport integral**
  - Creează PDF pentru **toți clienții** după filtrările curente (inclusiv scadențe/agenți).
  - Titlu adaptiv: *RAPORT FACTURI CLIENTI MULTIPLI* sau *RAPORT FACTURI – AGENT: Nume*.
3. **Raport individual (PDF per client)**
  - Selector dedicat de client + buton „Salvează Raport”.
  - Tabel cu **Nr. Crt.**, **Nr. Factură** (din Serie), **Data Factură**, **Data Scadență**, **Status Scadență**, **Sumă (RON)**, plus **TOTAL RESTANTE** și **TOTAL GENERAL**.

### Tehnic PDF

Framework: **ReportLab** (SimpleDocTemplate, Table, Paragraph, Image).

Pagina: **A4**, margini coerente, linii de separare (HR) și stiluri consistente.

Data/ora: generate în fusul orar **Europe/Bucharest**.

---

## 5. Logica de calcul & status scadență

- $\text{zile\_intarziere} = (\text{data\_curentă} - \text{DataScadenta}).\text{days}$ 
    - $> 0 \rightarrow \text{RESTANTĂ (x zile)}$
    - $= 0 \rightarrow \text{SCADENTĂ AZI}$
    - $< 0 \rightarrow \text{ÎN TERMEN (x zile)}$
  - Totaluri:
    - **TOTAL RESTANTE** – sumă Sold pentru rândurile cu întârziere  $> 0$ .
    - **TOTAL GENERAL** – sumă Sold pentru toate rândurile din raportul curent.
- 

## 6. Elemente UI & UX

- **Carduri KPI** curate (fără emoticoane), culori: roșu pentru solduri/atenționări.
  - **Data editor** cu checkbox *Selectat* → flux intuitiv pentru compunerea raportului multi-client.
  - **Previzualizare PDF + buton Descărcare** (link *data:application/pdf...*) cu denumire de fișier pe bază de timestamp.
  - Mesaje explicite când lipsesc date sau când filtrările duc la set vid.
-



## 7. Validări „edge cases”

- Conversii Data/DataScadenta cu errors='coerce' — valorile invalide sunt ignorate la status scadență.
  - **Număr factură**: se folosește din Serie; dacă lipsește, se generează **provizoriu** (F####) doar ca fallback în PDF.
  - **Excluderi**: afectează atât KPI-urile, cât și setul din care se compun rapoartele.
  - Toate agregările verifică existența coloanelor (Client, Sold, Agent, DataScadenta, Serie) pentru a evita erori.
- 

## 8. Rolul paginii în arhitectură

Modulul **Facturi Neîncasate** este instrumentul principal pentru managementul creanțelor:

- monitorizarea riscului de neîncasare,
  - prioritizarea acțiunilor de colectare,
  - generarea de rapoarte **profesionale** (individuale / multi-client) pentru comunicare cu clienții sau cu **agenții** de vânzări.
- 

## 9. Concluzii

Versiunea paginii **Facturi Neîncasate**:

- păstrează toate coloanele necesare în fluxul de lucru,
- afișează **numărul real al facturii** din Serie,
- oferă **rapoarte PDF** robuste (individuale și multi-client), cu statusuri de scadență clare și totaluri executive.

Modulul este proiectat pentru utilizare zilnică și integrare directă în procesele de **colectare creanțe** și **relații cu clienții**.

# Pagina Facturi Neachitate – Brenado For House

# 1. Scopul paginii

Modulul **Facturi Neachitate** centralizează datoriile către furnizori (cu sau fără efecte), oferind:

- vizualizarea sumelor restante și a scadențelor pe fiecare furnizor,
- filtre rapide pe perioade și scadență,
- **rapoarte PDF** continue (multi-furnizor) cu **TOTAL GENERAL pe raport**, gata de arhivare sau transmitere.

Versiunea curentă are un **header unic** în PDF – „*FACTURI NEACHITATE*” – iar **numele furnizorului apare în corp**, deasupra tabelului său. Raportul PDF este **continuu** (fără PageBreak între furnizori).

---

## 2. Structura logică a modulului

Fișier: **pages/facturi\_neachitate.py**

1. **Protecție acces**  
st.session\_state['page\_key'] = 'neachitate' + verificare\_autentificare().
  2. **Încărcare date & comutator sursă**
    - Bifa „**Cu Efecte**” determină sursa:
      - load\_neachitate() → **Cu efecte** (implicit)
      - load\_neachitate\_total() → **Total** (fără efecte)
    - Conversie date: Data, DataScadenta → datetime (cu errors='coerce').
  3. **KPI principal**
    - **Sold** total (RON), calculat din coloana Sold.
  4. **Filtrare & sinteză**
    - Filtre: **Furnizor**, **Scadență** („Azi” / „Săptămâna Curentă” / „Luna Curentă”), **până la data curentă**.
    - Grupare pe **Furnizor** cu agregare Total și Sold.
    - Editor interactiv cu coloană **Selectat** pentru compunerea listei de furnizori din raport.
  5. **Generator PDF**
    - \_generate\_multi\_supplier\_pdf(df, suppliers) → PDF continuu cu:
      - **Header unic** (logo + titlu + dată),
      - blocuri succesive (fără PageBreak) pentru fiecare furnizor selectat,
      - **TOTAL GENERAL RAPORT** la final.
- 

## 3. Surse de date și câmpuri utilizate

- Sursă: load\_neachitate() sau load\_neachitate\_total() (exporte ERP în XLSX).
- Câmpuri uzuale: Furnizor, Data, DataScadenta, Serie/Numar, Total, Sold, Valuta, PL etc.
- Numărul facturii este extras dintr-un set flexibil de coloane posibile (Serie, Numar, NumarFactura etc.); dacă lipsește, se folosește un identificator provizoriu în tabelul PDF.

---

## 4. Funcționalități cheie

### 4.1. Comutator „Cu Efecte”

- **Bifat** → vizualizează setul *cu efecte*.
- **Debifat** → vizualizează setul *total* (fără efecte).
- Toate metricile și rapoartele se refac în funcție de selecție.

### 4.2. KPI

- **Sold** total (RON) pentru selecția curentă.

### 4.3. Filtre operative

- **Furnizor** (multiselect).
- **Scadență**:
  - *Azi* – facturile scadente în data curentă,
  - *Săptămâna Curentă* – luni–duminică,
  - *Luna Curentă* – 1 → ultima zi a lunii curente.
- „**Data scadență până la DD/MM/YYYY**” – restrânge la toate facturile cu scadența ≤ astăzi.

### 4.4. Sintează pe Furnizor + selector de includere

- Tabel agregat (pe furnizor) cu coloanele **Total** și **Sold**.
- Coloană **Selectat** (checkbox) pentru a include furnizorii în raportul PDF.
- Două acțiuni:
  - **Previzualizează PDF** – pentru furnizorii bifați,
  - **Raport pentru toți clienții** – construiește raport pentru **toți furnizorii** din filtrarea curentă.

---

## 5. Raportare PDF (continuu, multi-furnizor)

### 5.1. Header unic

- Logo companie (dacă fișierul există) + titlu „**FACTURI NEACHITATE**” + data generării.
- Linie de separare și spațiu de respiro.

### 5.2. Bloc furnizor (în corp, fără PageBreak)

Pentru fiecare furnizor selectat (în ordinea apariției în setul filtrat):

- **Titlul cu numele furnizorului** (bold).
- **Tabel** cu coloane:
  - *Nr. Crt., 2) Nr. Factura, 3) Data Factura, 4) Data Scadenta, 5) Status Scadenta, 6) Suma (RON).*
- **Linii de total** pe furnizor:
  - **TOTAL RESTANTE** – sumă a rândurilor cu scadența depășită,
  - **TOTAL GENERAL** – sumă a tuturor rândurilor furnizorului.

#### Notă PDF:

- Headerul tabelului se repetă automat pe paginile următoare (*repeatRows=1*).
- Nu se inserează **PageBreak** între furnizori – raportul rămâne **continuu**.

### 5.3. Total pe întregul raport

- La final, secțiune cu „**TOTAL GENERAL RAPORT**” (RON), calculat ca suma Sold pentru toți furnizorii incluși.

## 6. Logica de scadență și status

- $zile = (data\_curentă - DataScadenta).days$ 
  - $> 0 \rightarrow$  **RESTANTĂ (x zile)**
  - $= 0 \rightarrow$  **SCADENȚĂ AZI**
  - $< 0 \rightarrow$  **ÎN TERMEN (x zile)**

În tabelul PDF, fiecare factură primește statusul corespunzător; la totaluri, **TOTAL RESTANTE** adună exclusiv rândurile cu zile  $> 0$ .

## 7. Elemente UI & UX

- **Card KPI** simplu (Sold total) cu evidențiere în roșu.
- **Editor interactiv** pentru sinteză furnizori, cu checkbox **Selectat** (selectare multiplă).
- **Previzualizare PDF** (dacă mediul suportă st.pdf) și **download** prin link direct (data:application/pdf...) cu denumiri de fișier datate.
- Mesaje clare pentru seturi vide sau combinații de filtre fără rezultate.

## 8. Validări & robustețe

- Conversii de dată cu errors='coerce' (valorile invalide sunt ignorate în calculul statusului).
- Extrage **Nr. Factură** dintr-o listă de coloane posibile; dacă lipsește, se folosește un fallback (F###) doar în PDF.

- Agregările verifică existența coloanelor (Furnizor, Sold, Total, Data, DataScadenta) pentru a evita erorile.
  - Ordinea furnizorilor în PDF urmează ordinea din setul filtrat (consistență cu UI).
- 

## 9. Concluzii

Modulul **Facturi Neachitate** oferă un flux complet pentru monitorizarea datoriilor către furnizori:

- comutare rapidă **Cu efecte/Total**,
- filtre operaționale pe **scadență**,
- sinteză **pe furnizor** cu selecție pentru raport,
- **PDF continuu** cu **TOTAL GENERAL** la final.

Acest design facilitează raportarea lunară/săptămânală și dialogul transparent cu furnizorii, menținând în același timp o trasabilitate clară a scadențelor.

# Pagina Scadențe Plăți cu Efecte – Brenado For House

## 1. Scopul paginii

Modulul **Scadențe Plăți cu Efecte** centralizează toate obligațiile de plată pe termen scurt (efecte comerciale, cambii, BO etc.), oferind:

- vizualizare rapidă a scadențelor pe zile și terți,
  - filtre flexibile (data scadenței, terț),
  - metrice cheie și **vizualizări** pentru planificarea cash-flow-ului (distribuție zilnică, heatmap, top terți, timeline).
- 

## 2. Localizare și protecție acces

Fișier: **pages/scadente\_plati.py**

Protecție: `st.session_state['page_key']='scadente' + verificare_autentificare()`.

---

### 3. Surse de date și câmpuri necesare

Sursă: `utils.data_loaders.load_scadente_plati()` (export ERP în XLSX).

Câmpuri utilizate (dacă există în fișier):

- **DataScadenta** (*datetime, obligatoriu pentru grafice*)
- **Data, DataEmiterii** (*datetime, opționale*)
- **Emitent, Tert** (*texte*)
- **Suma, Avans** (*numerice*)
- **StareLaData** (*text*)

Conversii de dată: DataScadenta, Data, DataEmiterii → datetime + coloane formatate pentru afișare (...\_Display, DataScadenta\_Formatata).

---

### 4. Indicatori principali (KPI)

- **Total**: suma câmpului **Suma** (RON) pentru setul curent de date, afișat într-un card KPI.
- 

### 5. Filtrare date

#### 5.1. Data scadenței (multiselect)

- Opțiuni prezentate ca **DD/MM/YYYY**, sortate cronologic (din DataScadenta).
- **Logică**: dacă sunt alese mai multe date, se ia **cea mai târzie** și se filtrează **toate scadențele ≤ data maximă selectată** (construiește o „prag-limită” comodă pentru utilizator).

#### 5.2. Tert (multiselect)

- Filtru după **Tert**; permite selectarea simultană a mai multor terți.

Aplicarea filtrelor se face incremental (întâi dată, apoi tert).

---

### 6. Tabel de lucru

Coloane afișate (unde există):

**Data Scadența, Stare La Data, Data, Data Emiterii, Emitent, Terț, Suma, Avans.**

Există mapare și fallback automat pentru coloanele de dată formatate (ex. dacă lipsesc coloanele ...\_Display, se folosește coloana brută).

Dacă lipsesc unele coloane, pagina anunță explicit și afișează toate coloanele disponibile din dataset (evitare erori + transparență).

**Statistici filtrate:** sub tabel, se calculează **Suma Filtrată** (RON) atunci când există filtre active.

---

## 7. Vizualizări grafice

Selecție perioadă (radio): **Săptămâna curentă / Luna curentă / Anul curent**

Filtrarea pentru vizualizări se face pe DataScadenta în funcție de opțiunea selectată.

### 7.1. Distribuție pe Zile (bar stacked)

- Grupare după **DataScadenta** și **Tert**; ordonare cronologică.
- Afișare ca **bară stivuită** (x = zi, y = sumă), culori pe terți.
- Hover cu format **RON**; legendă activă.

### 7.2. Heatmap Calendar (ziua lunii × luna)

- Agregare zilnică a sumei pe lună/zi (pivotal).
- **Heatmap** cu etichete de valori (rotunjite).
- Permite identificarea rapidă a zilelor cu vârfuri ale scadențelor.

### 7.3. Distribuție Terți (donut + tabel)

- **Top 10 terți** după sumă în perioada selectată.
- Donut chart (ponderi) + tabel compact cu **Suma** și **Procent**.

### 7.4. Timeline (bubble + cumulativ)

- **Bubble chart:** x = **DataScadenta**, y = **Tert**, mărimea = **Suma**.  
Categorii de sumă (legenda): < 5.000, 5.000–10.000, 10.000–20.000, > 20.000.
  - **Linie cumulativă** (sumă în timp) pentru aceeași perioadă – evoluția „dinamică” a obligațiilor.
- 

## 8. Comportamente și validări

- Conversii cu errors='coerce' (date invalide devin NaT și sunt ignorate corect la sortare/filtrare).

- Gestionarea **coloanelor lipsă** prin avertizare + afișarea setului disponibil.
  - Ordonări cronologice realizate pe coloanele **datetime** (nu pe stringuri), apoi se aplică formatarea.
  - Metrici și grafice rezistente la DataFrame gol (mesaje informative, fără erori).
- 

## 9. Beneficii operaționale

- **Planificare cash-flow:** evidențiază vârfurile pe zile și terți cheie.
  - **Prioritizare plăți:** heatmap + top terți facilitează ordonarea deciziilor.
  - **Control risc:** timeline-ul și cumulativul indică presiunea pe intervalul selectat.
- 

## 10. Concluzie

Pagina **Scadențe Plăți cu Efecte** oferă un tablou complet pentru monitorizarea obligațiilor pe termen scurt: filtrare rapidă, analiză vizuală detaliată și metrici sintetice. Designul robust gestionează date incomplete, păstrează ordinea temporală și pune în evidență atât „**cât**” (sume), cât și „**când**” (calendar), facilitând decizii de plată eficiente.