tactusoft / **loteriabogota**

Code Issues Pull requests Actions Projects Security and quality Insi

We are having a problem billing the locqum organization. Please update your payment method or call your payment provider for details on why the transaction failed.
You can always contact support with any questions.

Commit 2305c14

garciajavier385-sketch committed 2 weeks ago

```
intregracion comercial frontend
```

master (#1)

1 parent 5f7e86f commit 2305c14

7 files changed +1005 -543 lines changed

Top

Filter files...

- backend/src/main
  - java/co/gov/loteriabogota/loteriabackend/infrastructure/adapter
    - in/web/controller
      - ConnectionComercialSystemController.java
    - out/comercialSystem
      - ComercialSystemAdapter.java
  - resources/db/migration
    - V8__add_draw_number_and_extraordinario.sql
- frontend/src
  - app/features
    - payment
      - payment-result.component.ts
    - purchase
      - purchase.component.ts

environments

environment.prod.ts

environment.ts

7 files changed +1005 -543 lines changed

Search within code

…ller/ConnectionComercialSystemController.java

```
@@ -31,14 +31,14 @@ public ConnectionComercialSystemController(ComercialSystemUseCase comercialSyste
31 31          this.comercialSystemUseCase = comercialSystemUseCase;
32 32      }
33 33
34    -     @GetMapping("/series")
   34 +     @PostMapping("/series")
35 35      public ResponseEntity<SeriesResponse> getAvailableSeries(@Valid @RequestBody FindSeriesRequest request) throws KeyManagementException, NoSuchAlgorithmException, KeyStoreException {
36 36          SeriesResponse availableSeries = comercialSystemUseCase.findSeries(request);
37 37          return ResponseEntity.ok(availableSeries);
38 38
39 39      }
40 40
41    -     @GetMapping("/assignment")
   41 +     @PostMapping("/assignment")
42 42      public ResponseEntity<AssignmentNumberResponse> getAssignmentNumero(@Valid @RequestBody AssignmentNumberRequest request) throws KeyManagementException, NoSuchAlgorithmException, KeyStoreException {
43 43          AssignmentNumberResponse numberAssignment = comercialSystemUseCase.assignmentNumber(request);
44 44          return ResponseEntity.ok(numberAssignment);
@@ -60,5 +60,4 @@ public ResponseEntity<StatusPayCancelConfirmationResponse> payCancel(@Valid @Req
60 60
61 61      }
62 62
63    - }
64    -
   63 + }
```

…t/comercialSystem/ComercialSystemAdapter.java

Load Diff

Large diffs are not rendered by default.

…on/V8__add_draw_number_and_extraordinario.sql

```
@@ -11,7 +11,7 @@ UPDATE lotteries SET draw_date = '2026-02-19 21:00:00' WHERE id = 'a1b2c3d4-e5f6
11  11  INSERT INTO lotteries (id, name, description, image_url, price_per_fraction, total_fractions, draw_date, active, draw_number) VALUES
12  12  ('b2c3d4e5-f6a7-8901-bcde-f12345678901', 'Lotería de Bogotá Extraordinario',
13  13   'Sorteo extraordinario de la Lotería de Bogotá con premios especiales. No te pierdas esta oportunidad unica de ganar.',
14      -  '/images/loteria-bogota-extra.png', 7500, 4, '2026-03-12 21:00:00', true, 2835);
    14  +  '/images/loteria-bogota-extra.png', 7500, 4, '2026-03-12 21:00:00', true, 2820);
15  15
16  16  -- Seed available series for Extraordinario
17  17  INSERT INTO lottery_series (id, lottery_id, series_number) VALUES
@@ -22,4 +22,4 @@ INSERT INTO lottery_series (id, lottery_id, series_number) VALUES
22  22  ('22222222-2222-2222-2222-222222222205', 'b2c3d4e5-f6a7-8901-bcde-f12345678901', '104'),
23  23  ('22222222-2222-2222-2222-222222222206', 'b2c3d4e5-f6a7-8901-bcde-f12345678901', '150'),
24  24  ('22222222-2222-2222-2222-222222222207', 'b2c3d4e5-f6a7-8901-bcde-f12345678901', '151'),
25      - ('22222222-2222-2222-2222-222222222208', 'b2c3d4e5-f6a7-8901-bcde-f12345678901', '152');
    25  + ('22222222-2222-2222-2222-222222222208', 'b2c3d4e5-f6a7-8901-bcde-f12345678901', '152');
```

⊖

…/features/payment/payment-result.component.ts

```
@@ -5,6 +5,7 @@ import { environment } from '@env';
5   5   import { SpinnerComponent } from '@shared/components/spinner/spinner.component';
```

```
6   6     import { ErrorMessageComponent } from '@shared/components/error-message/error-
          message.component';
7   7     import { CurrencyPipe, DatePipe } from '@angular/common';
      8   + import { firstValueFrom } from 'rxjs';
8   9
9   10    interface PaymentStatus {
10  11      reference: string;
          @@ -28,6 +29,34 @@ interface PaymentStatus {
28  29      }[];
29  30    }
30  31
      32  + interface PendingCartItem {
      33  +   lotteryId: string;
      34  +   lotteryName: string;
      35  +   drawNumber: number | null;
      36  +   drawDate: string;
      37  +   number: string;
      38  +   series: string;
      39  +   fractions: number;
      40  +   price: number;
      41  +   reservationCode?: string | null;
      42  +   reservationAuditCode?: string | null;
      43  +   reservationMessage?: string | null;
      44  + }
      45  +
      46  + interface ComercialPayCancelItem {
      47  +   SORTEO: string;
      48  +   NUMERO: string;
      49  +   SERIE: string;
      50  + }
      51  +
      52  + interface ComercialPayCancelResponse {
      53  +   estado: string;
      54  +   codigo: string;
      55  +   codigoAuditoria: string | null;
      56  +   mensaje: string;
      57  +   data: ComercialPayCancelItem[];
      58  + }
      59  +
31  60    @Component({
32  61      selector: 'app-payment-result',
33  62      standalone: true,
```

```
@@ -159,6 +188,12 @@ export class PaymentResultComponent implements OnInit {
159 188     readonly error = signal<string | null>(null);
160 189     readonly result = signal<PaymentStatus | null>(null);
161 190
    191 +   private readonly comercialDistribuidor = environment.connectionComercial.distribuidor;
    192 +   private readonly comercialToken = this.generateComercialToken();
    193 +
    194 +   private static readonly PENDING_PAYMENT_CART_KEY = 'loteria_pending_payment_cart';
    195 +   private static readonly CANCELLED_REFERENCE_PREFIX = 'loteria_cancelled_reference_';
    196 +
162 197     ngOnInit(): void {
163 198       const reference = this.route.snapshot.queryParamMap.get('reference');
164 199       if (!reference) {
@@ -169,14 +204,122 @@ export class PaymentResultComponent implements OnInit {
169 204
170 205       this.http.get<PaymentStatus>(`${environment.apiUrl}/payments/${reference}/status`)
171 206         .subscribe({
172     -         next: (res) => {
    207 +         next: async (res) => {
173 208             this.result.set(res);
174     -           this.loading.set(false);
    209 +
    210 +           try {
    211 +             if (res.status === 'REJECTED') {
    212 +               await this.releasePendingReservations(reference);
    213 +             } else if (res.status === 'APPROVED') {
    214 +               this.clearPendingCart(reference);
    215 +             }
    216 +           } catch (e: any) {
    217 +             this.error.set(e?.message || 'El pago fue rechazado pero no se pudieron liberar todas las reservas');
    218 +           } finally {
    219 +             this.loading.set(false);
    220 +           }
175 221           },
176 222           error: (err) => {
```

```
177  223              this.error.set(err.error?.message ?? 'Error al consultar el estado del pago');
178  224              this.loading.set(false);
179  225            },
180  226          });
181  227      }
     228  +
     229  +   private async releasePendingReservations(reference: string): Promise<void> {
     230  +     const alreadyCancelled = localStorage.getItem(
     231  +       PaymentResultComponent.CANCELLED_REFERENCE_PREFIX + reference
     232  +     );
     233  +
     234  +     if (alreadyCancelled === '1') {
     235  +       return;
     236  +     }
     237  +
     238  +     const raw = localStorage.getItem(PaymentResultComponent.PENDING_PAYMENT_CART_KEY);
     239  +     if (!raw) {
     240  +       return;
     241  +     }
     242  +
     243  +     let items: PendingCartItem[] = [];
     244  +     try {
     245  +       items = JSON.parse(raw) as PendingCartItem[];
     246  +     } catch {
     247  +       localStorage.removeItem(PaymentResultComponent.PENDING_PAYMENT_CART_KEY);
     248  +       return;
     249  +     }
     250  +
     251  +     if (!items.length) {
     252  +       localStorage.removeItem(PaymentResultComponent.PENDING_PAYMENT_CART_KEY);
     253  +       return;
     254  +     }
     255  +
     256  +     const results = await Promise.allSettled(
     257  +       items.map(item => this.cancelReservedItem(item))
     258  +     );
     259  +
     260  +     const failed = results.flatMap((result, index) => {
     261  +       if (result.status === 'fulfilled') {
     262  +         return [];
```

```
+       }
+
+       const item = items[index];
+       const reason = result.reason?.message || 'No fue posible cancelar';
+       return [`${item.number}-${item.series}: ${reason}`];
+     });
+
+     if (failed.length > 0) {
+       throw new Error('Algunas reservas no pudieron cancelarse: ' + failed.join(' | '));
+     }
+
+     localStorage.setItem(
+       PaymentResultComponent.CANCELLED_REFERENCE_PREFIX + reference,
+       '1'
+     );
+     localStorage.removeItem(PaymentResultComponent.PENDING_PAYMENT_CART_KEY);
+   }
+
+   private clearPendingCart(reference: string): void {
+     localStorage.removeItem(PaymentResultComponent.PENDING_PAYMENT_CART_KEY);
+     localStorage.removeItem(
+       PaymentResultComponent.CANCELLED_REFERENCE_PREFIX + reference
+     );
+   }
+
+   private async cancelReservedItem(item: PendingCartItem): Promise<ComercialPayCancelResponse> {
+     if (!item.drawNumber) {
+       throw new Error('El item no tiene numero de sorteo');
+     }
+
+     const payload = {
+       distribuidor: this.comercialDistribuidor,
+       numero: item.number,
+       serie: item.series,
+       sorteo: String(item.drawNumber),
+       token: this.comercialToken,
+     };
+
+     const resp = await firstValueFrom(
+       this.http.post<ComercialPayCancelResponse>(
```

```
303 +          `${environment.apiUrl}/connectionComercial/payCancel`,
304 +          payload
305 +        )
306 +      );
307 +
308 +      if (resp.codigo === '200' && resp.mensaje === 'Cancelación realizada.') {
309 +        return resp;
310 +      }
311 +
312 +      throw new Error(resp.mensaje || 'No fue posible cancelar la reserva');
313 +    }
314 +    private generateComercialToken(): string {
315 +      const { distribuidor, nit, clave } = environment.connectionComercial;
316 +      const today = new Date();
317 +      const yyyy = today.getFullYear();
318 +      const mm = String(today.getMonth() + 1).padStart(2, '0');
319 +      const dd = String(today.getDate()).padStart(2, '0');
320 +      const fecha = `${yyyy}${mm}${dd}`;
321 +      const raw = `${distribuidor}|${nit}|${clave}|${fecha}`;
322 +      const base64 = btoa(raw);
323 +      return base64;
324 +    }
182 325   }
```

…c/app/features/purchase/purchase.component.ts

Load Diff

Large diffs are not rendered by default.

frontend/src/environments/environment.prod.ts

```
    ...   @@ -1,4 +1,9 @@
1   1     export const environment = {
2   2       production: true,
3   3       apiUrl: '/api',
    4   +   connectionComercial: {
    5   +     distribuidor: 'GELSA',
    6   +     nit: '830111257-3',
    7   +     clave: 'gelsa2025'
```

```
      8   +   }
4     9     };
```

frontend/src/environments/environment.ts

```
      ...     @@ -1,4 +1,9 @@
1     1       export const environment = {
2           -   production: false,
      2   +   production: true,
3     3       apiUrl: '/api',
      4   +   connectionComercial: {
      5   +     distribuidor: 'GELSA',
      6   +     nit: '830111257-3',
      7   +     clave: 'gelsa2025'
      8   +   }
4     9     };
```

Comments 0

Lock conversation

Comment

Subscribe You're not receiving notifications from this thread.