

Check Point - Design System: Templates de Stories

Visão Geral do Design

Filosofia Visual

PRINCÍPIOS DE DESIGN:

1. PREMIUM, NÃO GENÉRICO

- Parece feito por designer, não por app
- Sem bordas coloridas óbvias
- Tipografia refinada
- Espaço negativo generoso

2. INSTAGRAM-NATIVE

- Cores que funcionam no feed
- Proporções corretas (9:16)
- Área livre para stickers do user
- Não compete com o conteúdo do user

3. COMPARTILHÁVEL

- As pessoas têm que ter ORGULHO de postar
- Parece status, não propaganda
- Minimalista mas impactante
- Personalizável

4. RECONHECÍVEL

- Identidade visual consistente
- Pessoas reconhecem "isso é Check Point"
- Mas não é invasivo

1. Especificações Técnicas

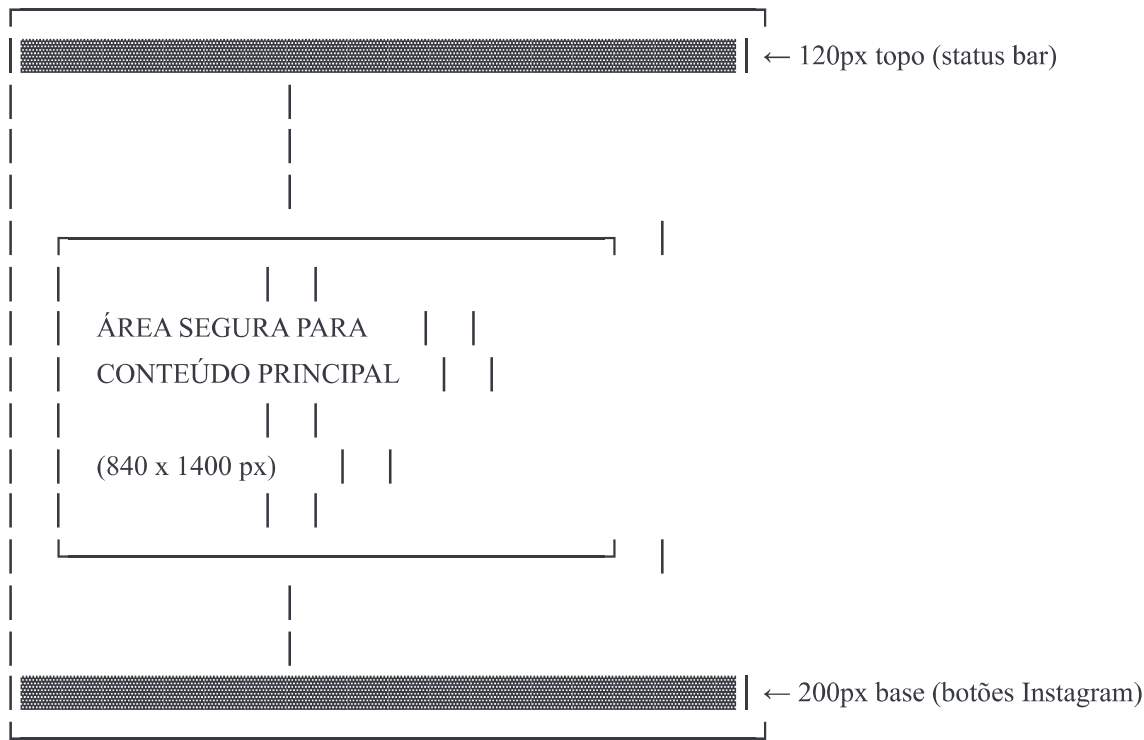
1.1 Dimensões e Formato

FORMATO BASE:

Dimensões: 1080 x 1920 pixels (9:16)
DPI: 72 (tela)

Formato export: PNG (estático) ou MP4 (animado)
Duração vídeo: 3-5 segundos (loop)

ÁREAS SEGURAS:



Margens laterais: 120px cada lado
Margem topo: 120px (status bar Instagram)
Margem base: 200px (reply bar, swipe up)

1.2 Paleta de Cores

CORES PRINCIPAIS:

TEMA DARK (Padrão):

- Background: #0A0A0A (quase preto)
- Surface: #1A1A1A (cards)
- Text Primary: #FFFFFF
- Text Secondary: #A0A0A0
- Accent: #7C3AED (roxo vibrante)
- Accent Alt: #10B981 (verde sucesso)

TEMA LIGHT:

- Background: #FAFAFA (off-white)
- Surface: #FFFFFF
- Text Primary: #1A1A1A
- Text Secondary: #6B7280
- Accent: #7C3AED

└─ Accent Alt: #10B981

VIP / GOLD:

└─ Gold Primary: #FFD700

└─ Gold Light: #FFF4CC

└─ Gold Dark: #B8860B

└─ Shimmer: Linear gradient 45deg
#FFD700 → #FFF4CC → #FFD700

NEON (Festas/Baladas):

└─ Background: #0D0D0D

└─ Neon Pink: #FF006E

└─ Neon Blue: #00D9FF

└─ Neon Purple: #BD00FF

└─ Glow effect: blur 20px, opacity 50%

GRADIENTS:

└─ Sunset: #FF6B6B → #FFA06B

└─ Ocean: #667EEA → #764BA2

└─ Forest: #11998E → #38EF7D

└─ Premium: #1A1A2E → #16213E

└─ Aurora: #7C3AED → #EC4899 → #F59E0B

1.3 Tipografia

FONTS:

PRINCIPAL: Inter (Google Fonts)

└─ Headlines: Inter Bold (700)

└─ Subheads: Inter SemiBold (600)

└─ Body: Inter Regular (400)

└─ Caption: Inter Medium (500)

ALTERNATIVA ELEGANTE: Playfair Display

└─ Para títulos VIP

└─ Serifa clássica

└─ Sensação premium

◀ HIERARQUIA: ▶

Tipo	Tamanho	Peso	Espaçamento	
Hero Title	72px	Bold	-2%	
Event Name	48px	Bold	-1%	

Section Title	32px	Semi	0%	
Body Large	24px	Reg	0%	
Body	20px	Reg	0%	
Caption	16px	Med	1%	
Micro	14px	Med	2%	

2. Template: Confirmei Presença

2.1 Layout Detalhado

TEMPLATE: CONFIRMEI PRESENÇA

Arquivo: template_confirmed.png / .mp4

Dimensões: 1080 x 1920 px

120px

Foto: 280x280px

[FOTO]Border-radius: 140px (círculo)

Border: 4px solid accent

Shadow: 0 20px 60px rgba(0,0,0,0.4)

@usernameFont: Inter Medium 24px

Color: text-secondary

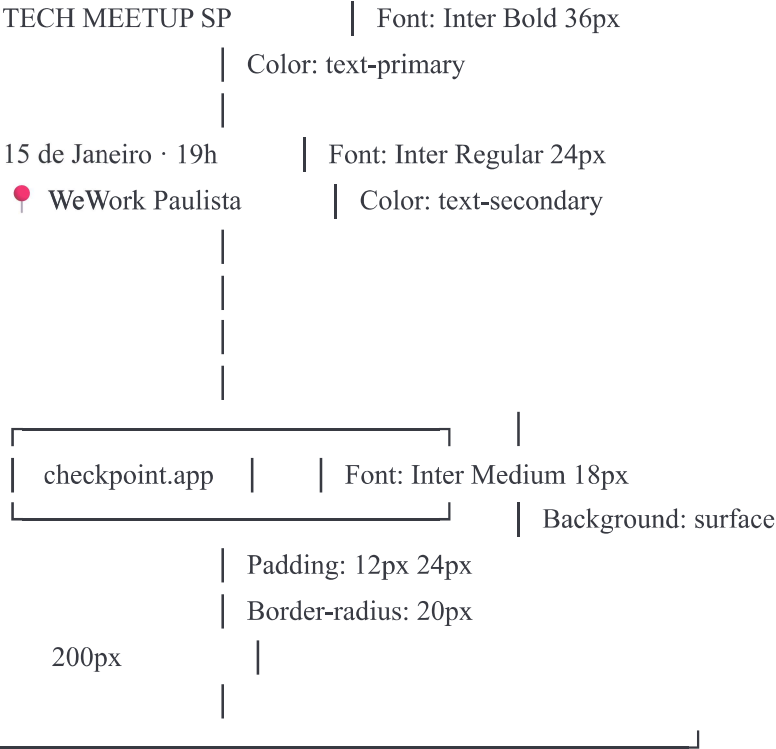
Linha: 2px, 200px width

Color: accent (30% opacity)

CONFIRMADOFont: Inter Bold 64px

✓Color: accent

Ícone: 48px, animate pulse



- ANIMAÇÃO (se vídeo):
- 0.0s - Fade in background
 - 0.2s - Foto scale in (0.8 → 1.0)
 - 0.4s - Username fade in
 - 0.6s - Linhas slide in dos lados
 - 0.8s - "CONFIRMADO" scale bounce
 - 1.0s - Checkmark aparece com pop
 - 1.2s - Info do evento fade in
 - 1.4s - CTA fade in
 - 3.0s - Loop (subtle pulse no checkmark)

2.2 Código de Geração (React)

jsx

```
// components/templates/ConfirmedTemplate.tsx
```

```
import React from 'react';
import { View, Text, Image, StyleSheet } from 'react-native';
import { LinearGradient } from 'expo-linear-gradient';

interface ConfirmedTemplateProps {
  userPhoto: string;
  username: string;
  eventName: string;
  eventDate: string;
  eventLocation: string;
  theme: 'dark' | 'light' | 'neon' | 'custom';
  customColors?: {
    background: string;
    accent: string;
    text: string;
  };
};

export const ConfirmedTemplate: React.FC<ConfirmedTemplateProps> = ({
  userPhoto,
  username,
  eventName,
  eventDate,
  eventLocation,
  theme = 'dark',
}) => {
  const colors = getThemeColors(theme);

  return (
    <View style={[styles.container, { backgroundColor: colors.background }]}>
      { /* User Photo */ }
      <View style={styles.photoContainer}>
        <Image
          source={{ uri: userPhoto }}
          style={[styles.photo, { borderColor: colors.accent }]}
        />
      </View>

      { /* Username */ }
      <Text style={[styles.username, { color: colors.textSecondary }]}>
        @{username}
      </Text>

      { /* Divider */ }
```

```

<View style={[styles.divider, { backgroundColor: colors.accent }]} />

{/* Confirmed Badge */}
<View style={styles.confirmedContainer}>
  <Text style={[styles.confirmedText, { color: colors.accent }]}>
    CONFIRMADO
  </Text>
  <Text style={styles.checkmark}>✓</Text>
</View>

{/* Divider */}
<View style={[styles.divider, { backgroundColor: colors.accent }]} />

{/* Event Info */}
<Text style={[styles.eventName, { color: colors.textPrimary }]}>
  {eventName}
</Text>
<Text style={[styles.eventDetails, { color: colors.textSecondary }]}>
  {eventDate}
</Text>
<Text style={[styles.eventDetails, { color: colors.textSecondary }]}>
  📍 {eventLocation}
</Text>

{/* CTA */}
<View style={[styles.cta, { backgroundColor: colors.surface }]}>
  <Text style={[styles.ctaText, { color: colors.textSecondary }]}>
    checkpoint.app
  </Text>
</View>
</View>
);
};

const styles = StyleSheet.create({
  container: {
    width: 1080,
    height: 1920,
    alignItems: 'center',
    justifyContent: 'center',
    paddingHorizontal: 120,
  },
  photoContainer: {
    marginBottom: 24,
  },
  photo: {
    width: 280,

```

```
    height: 280,
    borderRadius: 140,
    borderWidth: 4,
  },
  username: {
    fontSize: 24,
    fontWeight: '500',
    marginBottom: 48,
  },
  divider: {
    width: 200,
    height: 2,
    opacity: 0.3,
    marginVertical: 32,
  },
  confirmedContainer: {
    alignItems: 'center',
  },
  confirmedText: {
    fontSize: 64,
    fontWeight: '700',
    letterSpacing: -1,
  },
  checkmark: {
    fontSize: 48,
    marginTop: 16,
  },
  eventName: {
    fontSize: 36,
    fontWeight: '700',
    textAlign: 'center',
    marginTop: 48,
  },
  eventDetails: {
    fontSize: 24,
    marginTop: 12,
  },
  cta: {
    position: 'absolute',
    bottom: 240,
    paddingHorizontal: 24,
    paddingVertical: 12,
    borderRadius: 20,
  },
  ctaText: {
    fontSize: 18,
    fontWeight: '500',
```



```
},
});
```

3. Template: VIP

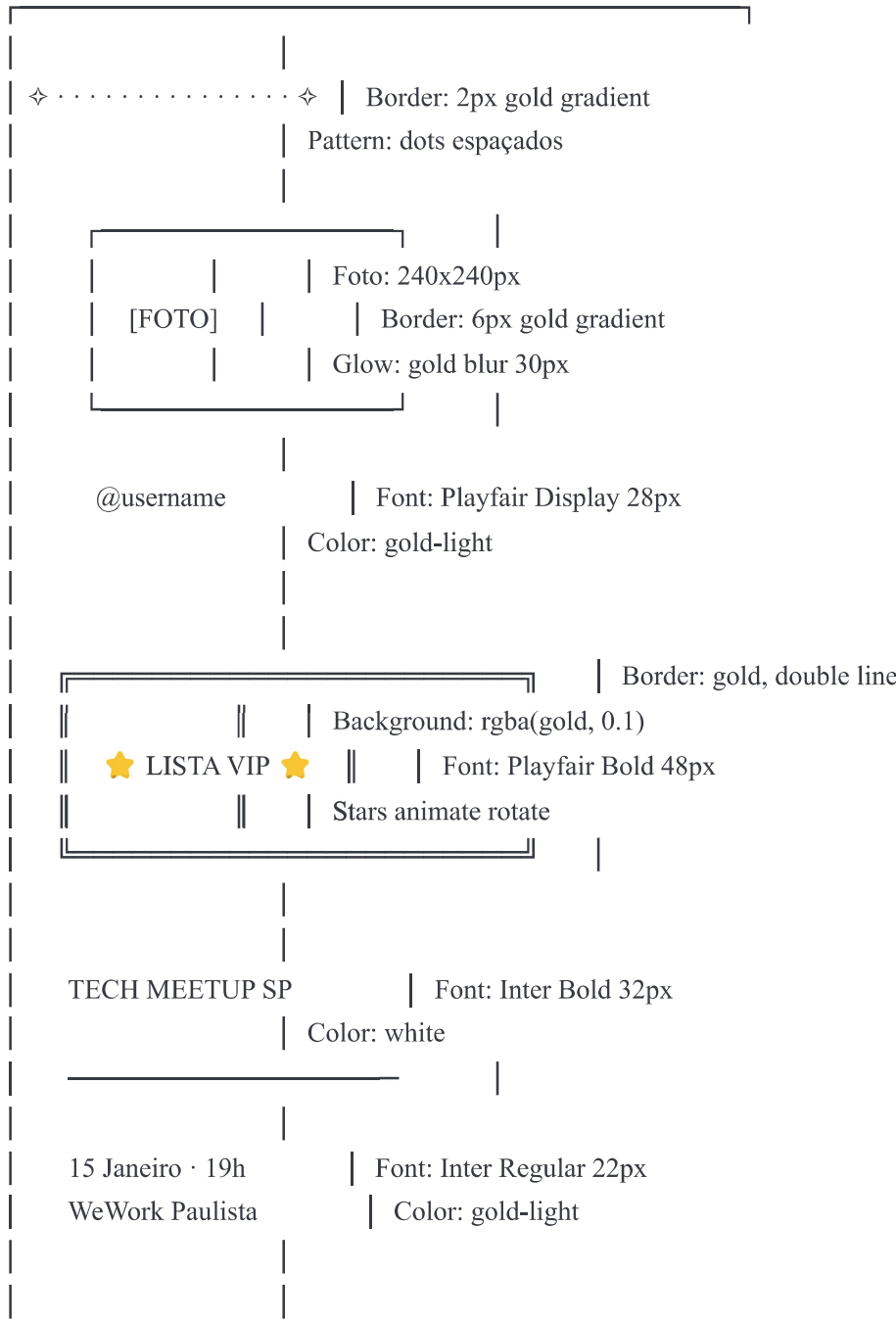
3.1 Layout Detalhado

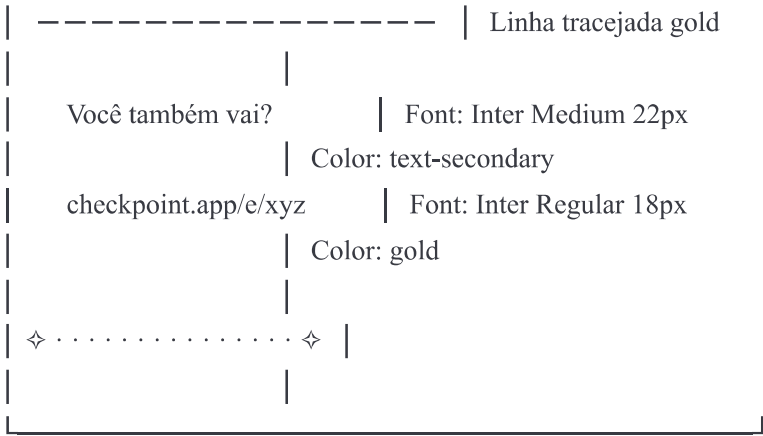
TEMPLATE: LISTA VIP

Arquivo: template_vip.png / .mp4

Dimensões: 1080 x 1920 px

Mood: Premium, exclusivo, elegante





EFEITOS ESPECIAIS:

- Shimmer effect na borda (animado)
- Stars (★) rotacionam lentamente
- Glow **dourado** na foto
- Partículas douradas sutis no fundo
- Textura de papel/convite sutil

ANIMAÇÃO (se vídeo):

- 0.0s - Fade in com partículas
- 0.3s - Borda dourada desenha (stroke animation)
- 0.6s - Foto aparece com glow
- 0.9s - Badge VIP scale in
- 1.2s - Stars começam a rotar
- 1.5s - Info fade in
- 2.0s - Shimmer passa pela borda
- Loop - Shimmer a cada 3s, stars rotating

3.2 CSS do Shimmer Effect

CSS

/ Shimmer effect para borda VIP */*

```
@keyframes shimmer {  
  0% {  
    background-position: -200% 0;  
  }  
  100% {  
    background-position: 200% 0;  
  }  
}  
  
.vip-border {  
  border: 2px solid transparent;  
  background:  
    linear-gradient(#0A0A0A, #0A0A0A) padding-box,  
    linear-gradient(  
      90deg,  
      #FFD700 0%,  
      #FFF4CC 25%,  
      #FFD700 50%,  
      #FFF4CC 75%,  
      #FFD700 100%  
    ) border-box;  
  background-size: 200% 100%;  
  animation: shimmer 3s ease-in-out infinite;  
}  
  
.vip-glow {  
  box-shadow:  
    0 0 30px rgba(255, 215, 0, 0.4),  
    0 0 60px rgba(255, 215, 0, 0.2),  
    0 0 90px rgba(255, 215, 0, 0.1);  
}  
  
@keyframes star-rotate {  
  from { transform: rotate(0deg); }  
  to { transform: rotate(360deg); }  
}  
  
.star {  
  animation: star-rotate 10s linear infinite;  
}
```

4. Template: Cheguei!

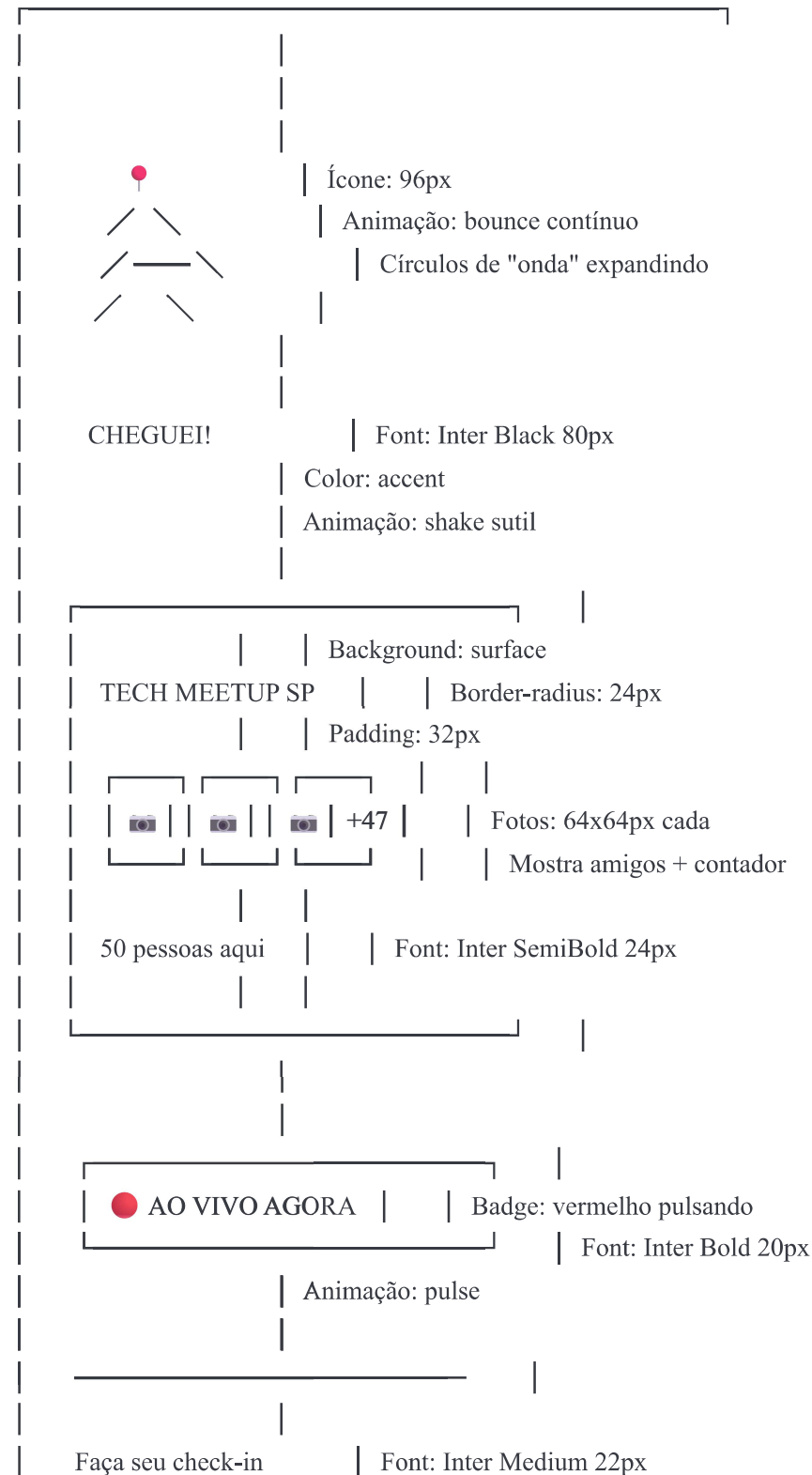
4.1 Layout Detalhado

TEMPLATE: CHEGUEI! (Check-in ao vivo)

Arquivo: template_arrived.png / .mp4

Dimensões: 1080 x 1920 px

Mood: Energético, ao vivo, FOMO



checkpoint.app

CTA call to action

ELEMENTOS DINÂMICOS:

Número de pessoas atualizado em tempo real

Fotos são de amigos do usuário que estão lá

Se nenhum amigo: mostra pessoas aleatórias

Badge "AO VIVO" pulsa a cada 1.5s

ANIMAÇÃO (se vídeo):

0.0s - Pin aparece de cima, bounce

0.3s - Ondas de radar expandem do pin

0.5s - "CHEGUEI!" shake + scale

0.8s - Card slide up

1.0s - Fotos aparecem uma a uma

1.2s - Counter incrementa

1.5s - Badge AO VIVO pisca

Loop - Ondas de radar, badge pulsando

5. Template: Wrap-up (Pós-evento)

5.1 Layout Detalhado

TEMPLATE: MEU CHECK POINT (Wrap-up)

Arquivo: template_wrapup.png / .mp4

Dimensões: 1080 x 1920 px

Mood: Celebração, conquista, Spotify Wrapped vibes

MEU CHECK POINT

Font: Inter Bold 28px

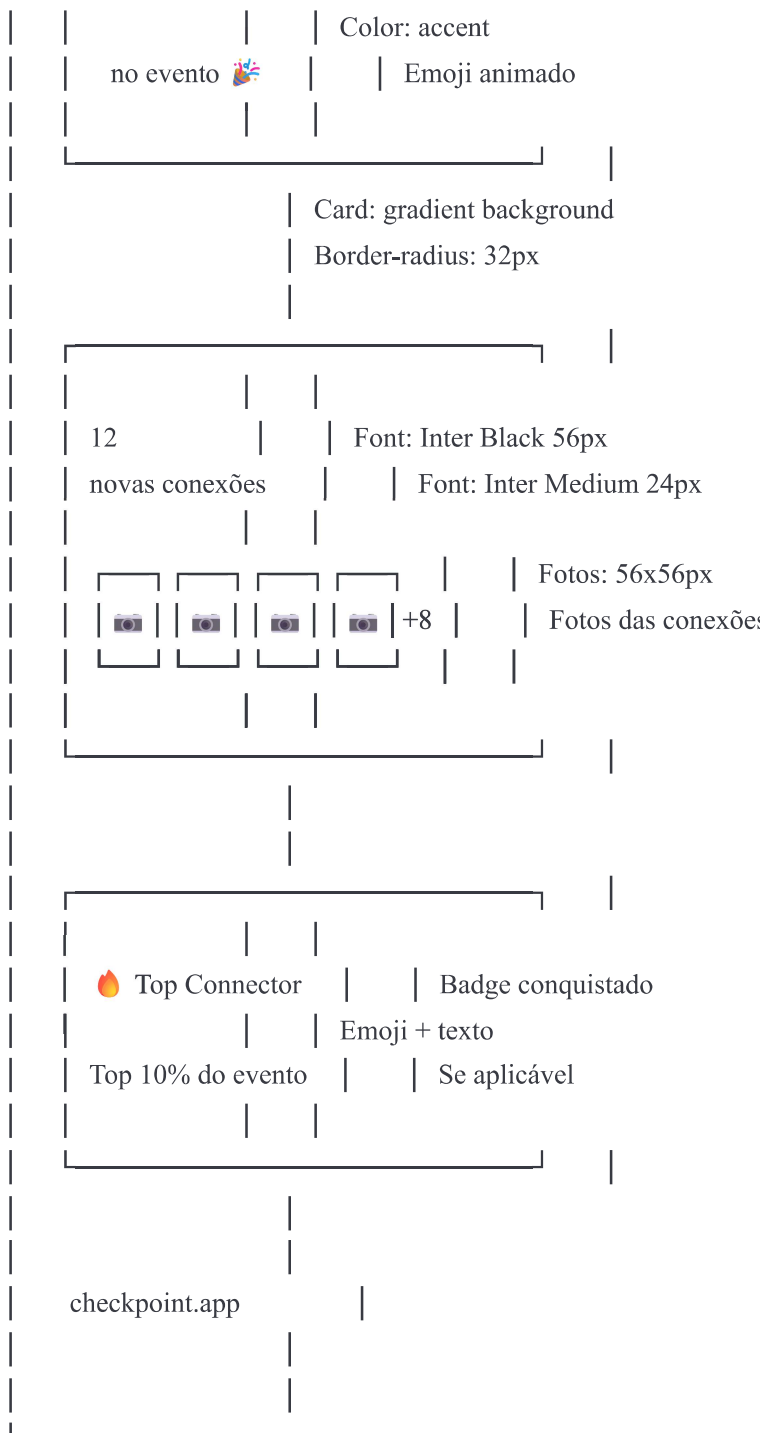
Color: text-secondary

TECH MEETUP SP

Uppercase, espaçado

3h 45m

Font: Inter Black 72px



VARIAÇÕES DE BADGE:

- 🔥 Top Connector - Top 10% em conexões
- ⚡ Early Bird - Chegou cedo
- ☀️ Party Starter - Primeiro check-in
- 🏆 Most Social - Mais interações
- 👑 VIP Experience - Era VIP
- 🎯 Focused - Ficou muito tempo

ANIMAÇÃO (se vídeo):


- 0.0s - Título fade in
- 0.3s - Card 1: números incrementam (count up)
- 0.6s - Emoji celebra (confetti)
- 1.0s - Card 2: fotos aparecem uma a uma
- 1.5s - Card 3: badge aparece com bounce


Amigos que vão


MENSAGEM PERSONALIZADA


Bora? 🔥

0/50 caracteres

 Stories

 Direct

 Salvar

 Copiar

COMPORTAMENTO:

- Preview atualiza em tempo real
- Tap no preview: abre fullscreen
- Toggle de opções: atualiza preview
- "Stories": gera imagem → abre Instagram
- "Direct": abre share sheet
- "Salvar": salva PNG na galeria
- "Copiar": copia deep link

7. Exportação e Compartilhamento

7.1 Fluxo Técnico

PROCESSO DE GERAÇÃO E COMPARTILHAMENTO:

1. RENDERIZAÇÃO

- Componente React renderiza template
- Canvas captura em 1080x1920
- Gera PNG (estático) ou MP4 (animado)
- Salva em cache temporário

2. COMPARTILHAMENTO INSTAGRAM STORIES

```
````javascript
import Share from 'react-native-share';

const shareToInstagramStories = async (imageUri) => {
 try {
 // Verifica se Instagram está instalado
 const isInstalled = await Linking.canOpenURL('instagram://');

 if (isInstalled) {
 const shareOptions = {
 social: Share.Social.INSTAGRAM_STORIES,
 backgroundImage: imageUri,
 stickerImage: CHECKPOINT_LOGO_URI,
 backgroundBottomColor: '#0A0A0A',
 backgroundTopColor: '#0A0A0A',
 attributionURL: 'https://checkpoint.app',
 };

 await Share.shareSingle(shareOptions);

 // Track analytics
 analytics.track('share_completed', {
 platform: 'instagram_stories',
 template_type: templateType,
 });
 } else {
 // Fallback: salvar e instruir
 await saveToGallery(imageUri);
 showToast('Imagem salva! Abra o Instagram e adicione aos Stories');
 }
 } catch (error) {
 console.error('Share error:', error);
 }
};
````
```

3. COMPARTILHAMENTO GENÉRICO

```
````javascript
const shareGeneric = async (imageUri, eventUrl) => {
 const shareOptions = {
 title: 'Check Point',
 message: `Confirmei presença! ${eventUrl}`,
 url: imageUri,
 type: 'image/png',
 };
};
```

```
 await Share.open(shareOptions);
 };
 ...
```

#### 4. SALVAR NA GALERIA

```
```javascript
import CameraRoll from '@react-native-camera-roll/camera-roll';

const saveToGallery = async (imageUri) => {
  await CameraRoll.save(imageUri, { type: 'photo' });
  showToast('Salvo na galeria! 📷');
};
...`
```

5. DEEP LINK NO CARD

- └─ URL: `checkpoint.app/e/{event_slug}`
- └─ Detecta se app instalado
- └─ Se sim: abre app direto no evento
- └─ Se não: abre App Store / Play Store

8. Biblioteca de Ícones e Assets

ASSETS NECESSÁRIOS:

ÍCONES (SVG):

- └─ `pin.svg` - 📍 Location pin
- └─ `star.svg` - ⭐ VIP star
- └─ `check.svg` - ✓ Checkmark
- └─ `fire.svg` - 🔥 Fire/trending
- └─ `sparkles.svg` - ✨ Sparkles
- └─ `crown.svg` - 👑 Crown/VIP
- └─ `live.svg` - 🟠 Live indicator
- └─ `logo.svg` - Check Point logo

BADGES (PNG com transparência):

- └─ `badge_vip.png` - Badge VIP dourado
- └─ `badge_confirmed.png` - Badge confirmado
- └─ `badge_present.png` - Badge presente
- └─ `badge_connector.png` - Badge top connector
- └─ `badge_early_bird.png` - Badge early bird
- └─ `badge_host.png` - Badge organizador

TEXTURAS:

- └─ noise_subtle.png - Textura de ruído sutil
- └─ paper_texture.png - Textura de papel (VIP)
- └─ gradient_aurora.png - Gradient aurora
- └─ confetti_particles.png - Partículas de confetti

ANIMAÇÕES (Lottie JSON):

- └─ checkmark_success.json - Checkmark animado
- └─ confetti_burst.json - Explosão de confetti
- └─ radar_pulse.json - Pulso de radar
- └─ shimmer.json - Efeito shimmer
- └─ star_spin.json - Estrela girando
- └─ live_pulse.json - Indicador ao vivo

FONTES:

- └─ Inter-Regular.ttf
- └─ Inter-Medium.ttf
- └─ Inter-SemiBold.ttf
- └─ Inter-Bold.ttf
- └─ Inter-Black.ttf
- └─ PlayfairDisplay-Regular.ttf
- └─ PlayfairDisplay-Bold.ttf

9. Checklist de Implementação

DESIGN SYSTEM - TEMPLATES:

SETUP:

- ☐ Configurar canvas rendering (react-native-view-shot)
- ☐ Instalar share library (react-native-share)
- ☐ Configurar deep linking
- ☐ Importar fontes (Inter, Playfair)
- ☐ Criar asset library (ícones, texturas)

TEMPLATES ESTÁTICOS:

- ☐ Template "Confirmei" - Dark
- ☐ Template "Confirmei" - Light
- ☐ Template "VIP" - Gold
- ☐ Template "Cheguei" - Dark
- ☐ Template "Cheguei" - Neon
- ☒ Template "Wrap-up" - Dark
- ☐ Template "Wrap-up" - Gradient

TEMPLATES ANIMADOS (V1.1):

- ☐ Animação checkmark success
- ☐ Animação shimmer VIP
- ☐ Animação radar pulse
- ☐ Animação confetti celebration
- ☐ Export como MP4/GIF

TELA DE PERSONALIZAÇÃO:

- ☐ Preview em tempo real
- ☐ Seletor de tema
- ☐ Toggles de elementos
- ☐ Campo de mensagem
- ☐ Botões de compartilhamento

INTEGRAÇÕES:

- ☐ Share para Instagram Stories
- ☐ Share para Instagram Direct
- ☐ Share genérico (WhatsApp, etc)
- ☐ Salvar na galeria
- ☐ Copiar link

ANALYTICS:

- ☐ Track share_started
- ☐ Track share_completed
- ☐ Track share_cancelled
- ☐ Track template_type
- ☐ Track destination_platform