💻 LA RESERVA - DESARROLLO WEB

PARTE 4-1: Componentes UI (Parte 1)

Versión: 1.0

Fecha: Octubre 2025

Tiempo de lectura: 20 minutos

📑 CONTENIDO

1. COMPONENTES UI BASE

Los componentes UI base son reutilizables en todo el sitio. Se construyen con Tailwind y siguen el design system de La Reserva.

Estructura de archivos:

```
src/components/ui/
├── Button.astro          # Botones
├── Card.astro            # Cards
├── Input.astro           # Inputs de formulario
├── Badge.astro           # Badges de estado
├── Loading.astro         # Indicadores de carga
├── Modal.tsx             # Modales (React)
└── Toast.tsx             # Notificaciones (React)
```

2. BUTTON COMPONENT

2.1 Button.astro

```
---
// src/components/ui/Button.astro
import { cn } from '@/utils/utils';

interface Props {
  variant?: 'primary' | 'secondary' | 'outline' | 'ghost';
  size?: 'sm' | 'md' | 'lg';
  href?: string;
  type?: 'button' | 'submit' | 'reset';
  disabled?: boolean;
  fullWidth?: boolean;
  class?: string;
}

const {
  variant = 'primary',
  size = 'md',
  href,
  type = 'button',
  disabled = false,
  fullWidth = false,
  class: className,
} = Astro.props;

const baseClasses = 'inline-flex items-center justify-center font-semibol

const variantClasses = {
  primary: 'bg-gradient-to-r from-primary-500 to-accent-500 text-secondar
  secondary: 'bg-transparent border-2 border-primary-500 text-primary-50
  outline: 'bg-transparent border-2 border-white text-white hover:bg-whit
  ghost: 'bg-transparent text-primary-500 hover:bg-primary-50',
};

const sizeClasses = {
  sm: 'px-4 py-2 text-xs',
  md: 'px-6 py-3 text-sm md:text-base',
  lg: 'px-8 py-4 text-base md:text-lg',
};
```

```
const classes = cn(
  baseClasses,
  variantClasses[variant],
  sizeClasses[size],
  fullWidth && 'w-full',
  className
);

const Tag = href ? 'a' : 'button';
---

<Tag
  class={classes}
  href={href}
  type={!href ? type : undefined}
  disabled={!href ? disabled : undefined}
>
  <slot />
</Tag>
```

2.2 Ejemplos de uso

```
---
import Button from '@/components/ui/Button.astro';
---

<!-- Botón primario -->
<Button variant="primary">
  Cotizar Evento
</Button>

<!-- Botón secundario -->
<Button variant="secondary" size="lg">
  Ver Portafolio
</Button>

<!-- Botón con link -->
<Button variant="primary" href="/cotizacion">
  Solicitar Cotización
</Button>
```

```
<!-- Botón deshabilitado -->
<Button variant="primary" disabled>
  Procesando...
</Button>

<!-- Botón full width -->
<Button variant="primary" fullWidth>
  Enviar Formulario
</Button>

<!-- Botón con clases custom -->
<Button variant="ghost" class="mt-4">
  Cancelar
</Button>
```

---

3. CARD COMPONENT

3.1 Card.astro

```
---
// src/components/ui/Card.astro
import { cn } from '@/utils/utils';

interface Props {
  variant?: 'default' | 'bordered' | 'elevated';
  padding?: 'none' | 'sm' | 'md' | 'lg';
  hover?: boolean;
  class?: string;
}

const {
  variant = 'default',
  padding = 'md',
  hover = false,
  class: className,
} = Astro.props;

const baseClasses = 'bg-white rounded-lg overflow-hidden transition-all d

const variantClasses = {
  default: 'shadow-md',
  bordered: 'border-2 border-secondary-200',
  elevated: 'shadow-xl',
};

const paddingClasses = {
  none: '',
  sm: 'p-4',
  md: 'p-6',
  lg: 'p-8',
};

const classes = cn(
  baseClasses,
  variantClasses[variant],
  paddingClasses[padding],
  hover && 'hover:shadow-2xl hover:-translate-y-1',
  className
);
---

<div class={classes}>
  <slot />
</div>
```

3.2 Card con Header y Footer

```
---
// src/components/ui/CardWithHeader.astro
import { cn } from '@/utils/utils';

interface Props {
  title?: string;
  subtitle?: string;
  class?: string;
}

const { title, subtitle, class: className } = Astro.props;
---

<div class={cn('bg-white rounded-lg shadow-md overflow-hidden', className
  {(title || subtitle) && (
    <div class="px-6 py-4 border-b border-secondary-200">
```

```
      {title && (
        <h3 class="text-xl font-display font-semibold text-secondary-600"
          {title}
        </h3>
      )}
      {subtitle && (
        <p class="text-sm text-secondary-400 mt-1">
          {subtitle}
        </p>
      )}
    </div>
  )}

  <div class="p-6">
    <slot />
  </div>

  <slot name="footer" />
</div>
```

3.3 Ejemplos de uso

```
---
import Card from '@/components/ui/Card.astro';
import CardWithHeader from '@/components/ui/CardWithHeader.astro';
---

<!-- Card básico -->
<Card>
  <h3>Título del card</h3>
  <p>Contenido del card</p>
</Card>

<!-- Card con hover effect -->
<Card hover>
  <img src="/image.jpg" alt="..." class="mb-4" />
  <h3>Card con hover</h3>
  <p>Pasa el mouse sobre mí</p>
</Card>

<!-- Card sin padding (para imágenes) -->
<Card padding="none">
```

```
  <img src="/hero.jpg" alt="..." class="w-full" />
  <div class="p-6">
    <h3>Card con imagen</h3>
  </div>
</Card>

<!-- Card con header -->
<CardWithHeader
  title="Título del Card"
  subtitle="Subtítulo opcional"
>
  <p>Contenido del card</p>

  <div slot="footer" class="px-6 py-4 bg-secondary-50 border-t">
    <button>Acción</button>
  </div>
</CardWithHeader>
```

---

4. INPUT COMPONENTS

4.1 Input.astro

```
---
// src/components/ui/Input.astro
import { cn } from '@/utils/utils';

interface Props {
  type?: 'text' | 'email' | 'tel' | 'number' | 'password' | 'date' | 'tim
  name: string;
  id?: string;
  label?: string;
  placeholder?: string;
  value?: string | number;
  required?: boolean;
  disabled?: boolean;
  error?: string;
  hint?: string;
  class?: string;
}

const {
  type = 'text',
  name,
  id = name,
  label,
  placeholder,
  value,
  required = false,
  disabled = false,
  error,
  hint,
  class: className,
} = Astro.props;

const inputClasses = cn(
  'w-full px-4 py-3 border rounded-lg transition-all duration-200',
  'focus:outline-none focus:ring-2 focus:ring-primary-500 focus:border-tr
  'disabled:bg-secondary-100 disabled:cursor-not-allowed',
  error ? 'border-red-500' : 'border-secondary-200',
  className
);
---

<div class="space-y-2">
  {label && (
    <label for={id} class="block text-sm font-semibold text-secondary-600
      {label}
      {required && <span class="text-red-500 ml-1">*</span>}
    </label>
  )}

  <input
    type={type}
    name={name}
    id={id}
    class={inputClasses}
    placeholder={placeholder}
    value={value}
    required={required}
    disabled={disabled}
  />

  {error && (
    <p class="text-sm text-red-500">
      {error}
    </p>
  )}

  {hint && !error && (
    <p class="text-sm text-secondary-400">
      {hint}
    </p>
  )}
</div>
```

4.2 Textarea.astro

```
---
// src/components/ui/Textarea.astro
import { cn } from '@/utils/utils';

interface Props {
  name: string;
  id?: string;
  label?: string;
  placeholder?: string;
  value?: string;
  rows?: number;
  required?: boolean;
  disabled?: boolean;
  error?: string;
  hint?: string;
  class?: string;
}

const {
  name,
  id = name,
  label,
  placeholder,
  value,
  rows = 4,
  required = false,
  disabled = false,
  error,
  hint,
  class: className,
} = Astro.props;

const textareaClasses = cn(
  'w-full px-4 py-3 border rounded-lg transition-all duration-200 resize-
  'focus:outline-none focus:ring-2 focus:ring-primary-500 focus:border-tr
  'disabled:bg-secondary-100 disabled:cursor-not-allowed',
  error ? 'border-red-500' : 'border-secondary-200',
  className
);
---

<div class="space-y-2">
  {label && (
    <label for={id} class="block text-sm font-semibold text-secondary-60
      {label}
      {required && <span class="text-red-500 ml-1">*</span>}
    </label>
  )}

  <textarea
    name={name}
    id={id}
    class={textareaClasses}
    placeholder={placeholder}
    rows={rows}
    required={required}
    disabled={disabled}
  >{value}</textarea>

  {error && (
    <p class="text-sm text-red-500">
      {error}
    </p>
  )}

  {hint && !error && (
    <p class="text-sm text-secondary-400">
      {hint}
    </p>
  )}
</div>
```

4.3 Select.astro

```
---
// src/components/ui/Select.astro
import { cn } from '@/utils/utils';

interface Option {
  value: string;
  label: string;
}

interface Props {
  name: string;
  id?: string;
  label?: string;
  options: Option[];
  value?: string;
  required?: boolean;
  disabled?: boolean;
  error?: string;
  placeholder?: string;
  class?: string;
}

const {
  name,
  id = name,
  label,
  options,
  value,
  required = false,
  disabled = false,
  error,
  placeholder = 'Selecciona una opción',
  class: className,
} = Astro.props;

const selectClasses = cn(
  'w-full px-4 py-3 border rounded-lg transition-all duration-200',
  'focus:outline-none focus:ring-2 focus:ring-primary-500 focus:border-tr
  'disabled:bg-secondary-100 disabled:cursor-not-allowed',
  'appearance-none bg-no-repeat bg-right',
  error ? 'border-red-500' : 'border-secondary-200',
  className
);
---
```

```
<div class="space-y-2">
  {label && (
    <label for={id} class="block text-sm font-semibold text-secondary-600
      {label}
      {required && <span class="text-red-500 ml-1">*</span>}
    </label>
  )}

  <div class="relative">
    <select
      name={name}
      id={id}
      class={selectClasses}
      required={required}
      disabled={disabled}
    >
      <option value="" disabled selected={!value}>
        {placeholder}
      </option>
      {options.map(option => (
        <option value={option.value} selected={value === option.value}>
          {option.label}
        </option>
      ))}
    </select>

    <!-- Chevron icon -->
    <div class="absolute right-3 top-1/2 -translate-y-1/2 pointer-events-
      <svg class="w-5 h-5 text-secondary-400" fill="none" stroke="current
        <path stroke-linecap="round" stroke-linejoin="round" stroke-width
      </svg>
    </div>
  </div>

  {error && (
    <p class="text-sm text-red-500">
      {error}
    </p>
  )}
</div>
```

4.4 Ejemplos de uso

```
---
import Input from '@/components/ui/Input.astro';
import Textarea from '@/components/ui/Textarea.astro';
import Select from '@/components/ui/Select.astro';
import { EVENT_TYPES } from '@/utils/constants';
---

<form>
  <!-- Input básico -->
  <Input
    name="name"
    label="Nombre completo"
    placeholder="Juan Pérez"
    required
  />

  <!-- Input con error -->
  <Input
    name="email"
    type="email"
    label="Email"
    error="Email inválido"
  />

  <!-- Input con hint -->
  <Input
    name="phone"
    type="tel"
    label="Teléfono"
    hint="Formato: 999 888 777"
  />

  <!-- Textarea -->
  <Textarea
    name="message"
    label="Mensaje"
    placeholder="Cuéntanos sobre tu evento"
    rows={6}
  />

  <!-- Select -->
  <Select
```