import {

showToast,

showConfirm,

showConfirmOkOnly,

renderizarCabeceraContactos,

renderizarFilasContactos

} from './utils.js';

let contactosMisContactosCompleta = []; // todos los contactos filtrados

let paginaActualMisContactos = 1;

const PAGE\_SIZE\_MIS\_CONTACTOS = 10;

let terminoBusquedaMisContactos = "";

function renderizarMisContactos(contactos) {

contactosMisContactosCompleta = contactos;

paginaActualMisContactos = 1;

renderizarCabeceraMisContactos(contactos);

renderizarFilasMisContactos(); // sin parámetro

}

async function renderizarCabeceraMisContactos(contactos) {

const containerMC = document.getElementById("mis-contactos-list");

if (containerMC.querySelector(".contactos-header")) return; // Si la cabecera está renderizada, terminar. contactos-header es la clase con la que están creadas las columnas de la cabecera

const searchWrapper = document.createElement("div");

searchWrapper.style.margin = "0 auto 0.5em auto";

searchWrapper.style.display = "flex";

searchWrapper.style.alignItems = "center";

searchWrapper.style.gap = "0.5em";

searchWrapper.style.maxWidth = "440px";

const searchIcon = document.createElement("i");

searchIcon.className = "fas fa-search";

searchIcon.style.color = "white";

const searchInput = document.createElement("input");

searchInput.type = "text";

searchInput.placeholder = "Buscar contacto...";

searchInput.style.flex = "1";

searchInput.style.padding = "0.3em 0.5em";

searchInput.style.borderRadius = "0.3em";

searchInput.style.border = "1px solid #ccc";

searchInput.style.width = "100%";

searchInput.style.boxSizing = "border-box";

searchWrapper.appendChild(searchIcon);

searchWrapper.appendChild(searchInput);

containerMC.appendChild(searchWrapper);

const header = document.createElement("div");

header.classList.add("contactos-header");

header.style.display = "flex";

header.style.fontWeight = "bold";

header.style.color = "white";

header.style.gap = "1em";

header.style.marginBottom = "0.5em";

header.style.flexWrap = "wrap";

const columnas = [

"Nombre",

"Quiero que reciba mis mensajes",

"Quiero que me vea",

"Eliminar",

];

columnas.forEach((title, i) => {

const col = document.createElement("div");

col.textContent = title;

col.style.flex = i === 0 ? "2" : "1";

col.style.textAlign = "center";

col.style.minWidth = "0";

col.style.overflow = "hidden";

col.style.textOverflow = "ellipsis";

header.appendChild(col);

});

containerMC.appendChild(header);

if (!containerMC.querySelector("#mis-contactos-body")) {

// \*\*\*

const body = document.createElement("div");

body.id = "mis-contactos-body";

containerMC.appendChild(body);

}

searchInput.addEventListener("input", () => {

// Guarda el valor del cuadro de texto

terminoBusquedaMisContactos = searchInput.value.trim().toLowerCase();

const termino = searchInput.value.trim().toLowerCase();

const filtrados = termino

? contactos.filter((c) => c.nombre.toLowerCase().includes(termino))

: contactos;

contactosMisContactosCompleta = filtrados;

paginaActualMisContactos = 1;

renderizarFilasMisContactos();

});

}

function renderizarFilasMisContactos() {

const containerMC = document.getElementById("mis-contactos-list");

const body = document.getElementById("mis-contacts-body");

const rows = containerMC.querySelectorAll(".contacto-row");

rows.forEach((row) => row.remove()); // Borra filas renderizadas

const desde = (paginaActualMisContactos - 1) \* PAGE\_SIZE\_MIS\_CONTACTOS;

const hasta = paginaActualMisContactos \* PAGE\_SIZE\_MIS\_CONTACTOS;

const visibles = contactosMisContactosCompleta.slice(desde, hasta);

visibles.forEach((c) => {

const row = document.createElement("div");

row.classList.add("contacto-row");

row.style.display = "flex";

row.style.alignItems = "center";

row.style.marginBottom = "0.5em";

row.style.gap = "1em";

row.style.color = "white";

const nombreColMC = document.createElement("div");

nombreColMC.style.flex = "2";

// Línea del nombre

const nombreSpan = document.createElement("div");

nombreSpan.textContent = c.nombre;

nombreSpan.style.fontWeight = "bold";

// Línea del teléfono

const telefonoSpan = document.createElement("div");

telefonoSpan.textContent = c.telefono || "(sin teléfono)";

telefonoSpan.style.fontSize = "0.9em";

telefonoSpan.style.opacity = "0.8";

// Insertar ambas líneas en la columna

nombreColMC.appendChild(nombreSpan);

nombreColMC.appendChild(telefonoSpan);

const checkboxMensajesMC = document.createElement("input");

checkboxMensajesMC.type = "checkbox";

checkboxMensajesMC.name = "recibirMensajes";

checkboxMensajesMC.className = "checkbox-input";

checkboxMensajesMC.checked = c.mensajes !== false; // \*\*

const mensajesColMC = document.createElement("div");

mensajesColMC.appendChild(checkboxMensajesMC);

mensajesColMC.style.flex = "1";

mensajesColMC.style.textAlign = "center";

const checkboxVisibilidadMC = document.createElement("input");

checkboxVisibilidadMC.type = "checkbox";

checkboxVisibilidadMC.name = "verme";

checkboxVisibilidadMC.className = "checkbox-input";

checkboxVisibilidadMC.checked = c.visibilidad !== false;

const vermeColMC = document.createElement("div");

vermeColMC.appendChild(checkboxVisibilidadMC);

vermeColMC.style.flex = "1";

vermeColMC.style.textAlign = "center";

// Actualiza los valores de mensajes en la base

checkboxMensajesMC.addEventListener("change", () => {

actualizarContacto(

c.nombre,

checkboxMensajesMC.checked,

checkboxVisibilidadMC.checked

);

});

checkboxVisibilidadMC.addEventListener("change", () => {

actualizarContacto(

c.nombre,

checkboxMensajesMC.checked,

checkboxVisibilidadMC.checked

);

});

row.appendChild(nombreColMC);

row.appendChild(mensajesColMC);

row.appendChild(vermeColMC);

// Actualiza los valores de visibilidad en la base

checkboxMensajesMC.addEventListener("change", () => {

actualizarContacto(

c.nombre,

checkboxMensajesMC.checked,

checkboxVisibilidadMC.checked

);

});

checkboxVisibilidadMC.addEventListener("change", () => {

actualizarContacto(

c.nombre,

checkboxMensajesMC.checked,

checkboxVisibilidadMC.checked

);

});

// 🗑️ Botón para eliminar contacto

const btnEliminar = document.createElement("button");

btnEliminar.innerHTML = `<i class="fas fa-trash-alt"></i>`;

btnEliminar.title = "Eliminar contacto";

btnEliminar.style.background = "none";

btnEliminar.style.border = "none";

btnEliminar.style.color = "white";

btnEliminar.style.cursor = "pointer";

btnEliminar.style.fontSize = "1.2em";

const eliminarCol = document.createElement("div");

eliminarCol.style.flex = "1";

eliminarCol.style.textAlign = "center";

eliminarCol.appendChild(btnEliminar);

btnEliminar.addEventListener("click", async () => {

const confirmar = await showConfirm(`¿Eliminar a ${c.nombre}?`);

if (!confirmar) return;

try {

const response = await fetch(

`${ROOT\_URL}/api/contacts?nombre=${encodeURIComponent(c.nombre)}`,

{

method: "DELETE",

headers: {

Authorization: `Bearer ${localStorage.getItem("token")}`,

},

}

);

if (!response.ok) {

const err = await response.json();

showToast("Error: " + err.error, "error");

return;

}

// Recargar la lista "Mis Contactos"

const res = await fetch(`${ROOT\_URL}/api/contacts`, {

headers: {

Authorization: `Bearer ${localStorage.getItem("token")}`,

},

});

const { contactos } = await res.json();

// 🔁 Refrescar lista de contactos de Google

if (typeof refrescarContactosGoogle === "function") {

await refrescarContactosGoogle(); // <- importante usar await para que se actualice a tiempo

}

// 🔁 Restaurar en contactosGoogleCargados

renderizarMisContactos(contactos, "mis-contactos-list");

} catch (err) {

console.error("Error al eliminar contacto:", err);

showToast("No se pudo eliminar el contacto", "error");

}

});

row.appendChild(eliminarCol);

const body = document.getElementById("mis-contacts-body");

(body || containerMC).appendChild(row);

renderizarControlesPaginadoMisContactos();

});

}

async function actualizarContacto(nombre, mensajes, visibilidad) {

try {

const response = await fetch(`${ROOT\_URL}/api/contacts`, {

method: "PUT",

headers: {

"Content-Type": "application/json",

Authorization: `Bearer ${localStorage.getItem("token")}`,

},

body: JSON.stringify({ nombre, mensajes, visibilidad }),

});

if (!response.ok) {

const err = await response.json();

showToast("Error al actualizar contacto: " + err.error, "error");

}

} catch (err) {

console.error("Error al actualizar contacto:", err);

showToast("No se pudo conectar con el servidor", "error");

}

}

function renderizarControlesPaginadoMisContactos() {

const container =

document.getElementById("mis-contacts-body") ||

document.getElementById("mis-contactos-list");

let paginador = document.getElementById("paginador-mis-contactos");

if (paginador) paginador.remove(); // borra si ya existe

const totalPaginas = Math.ceil(

contactosMisContactosCompleta.length / PAGE\_SIZE\_MIS\_CONTACTOS

);

if (totalPaginas <= 1) return;

paginador = document.createElement("div");

paginador.id = "paginador-mis-contactos";

paginador.style.textAlign = "center";

paginador.style.marginTop = "1em";

const crearBoton = (texto, habilitado, accion) => {

const btn = document.createElement("button");

btn.textContent = texto;

btn.disabled = !habilitado;

btn.style.margin = "0 0.3em";

btn.style.padding = "0.3em 0.7em";

btn.style.borderRadius = "0.3em";

btn.style.border = "none";

btn.style.cursor = habilitado ? "pointer" : "default";

btn.style.backgroundColor = habilitado ? "#007bff" : "#ccc";

btn.style.color = "white";

if (habilitado) btn.addEventListener("click", accion);

return btn;

};

paginador.appendChild(

crearBoton("⏮", paginaActualMisContactos > 1, () => {

paginaActualMisContactos = 1;

renderizarFilasMisContactos();

})

);

paginador.appendChild(

crearBoton("◀", paginaActualMisContactos > 1, () => {

paginaActualMisContactos--;

renderizarFilasMisContactos();

})

);

const input = document.createElement("input");

input.type = "number";

input.min = 1;

input.max = totalPaginas;

input.value = paginaActualMisContactos;

input.style.width = "40px";

input.style.textAlign = "center";

input.addEventListener("change", () => {

const nueva = parseInt(input.value);

if (!isNaN(nueva) && nueva >= 1 && nueva <= totalPaginas) {

paginaActualMisContactos = nueva;

renderizarFilasMisContactos();

} else {

input.value = paginaActualMisContactos;

}

});

const span = document.createElement("span");

span.textContent = ` / ${totalPaginas}`;

span.style.margin = "0 0.3em";

span.style.color = "white";

paginador.appendChild(input);

paginador.appendChild(span);

paginador.appendChild(

crearBoton("▶", paginaActualMisContactos < totalPaginas, () => {

paginaActualMisContactos++;

renderizarFilasMisContactos();

})

);

paginador.appendChild(

crearBoton("⏭", paginaActualMisContactos < totalPaginas, () => {

paginaActualMisContactos = totalPaginas;

renderizarFilasMisContactos();

})

);

container.appendChild(paginador);

}

window.addEventListener("DOMContentLoaded", async () => {

// 🔒 Verifica que esté logueado

try {

const response = await fetch(`${ROOT\_URL}/api/sessions/current`, {

method: "GET",

headers: {

Authorization: `Bearer ${localStorage.getItem("token")}`,

},

});

if (!response.ok) {

throw new Error("Token inválido");

}

} catch (err) {

window.location.href = "iniciarSesion.html";

return;

}

// 🟩 Toggle "Mis Contactos"

const misContactosList = document.getElementById("mis-contactos-list");

toggleMisContactos?.addEventListener("click", async () => {

const isVisible = misContactosList.style.display === "block";

misContactosList.style.display = isVisible ? "none" : "block"; // Togglea el estado de is Visible

// Rota el icono de chevron

const icon = toggleMisContactos.querySelector("i");

if (icon) {

icon.classList.toggle("rotate", !isVisible);

}

if (!isVisible) {

try {

const res = await fetch(`${ROOT\_URL}/api/contacts`, {

headers: {

Authorization: `Bearer ${localStorage.getItem("token")}`,

},

});

const { contactos } = await res.json();

// Filtra por el cuadro de texto de la lupita

const filtrados = terminoBusquedaMisContactos

? contactos.filter((c) =>

c.nombre.toLowerCase().includes(terminoBusquedaMisContactos)

)

: contactos;

renderizarMisContactos(filtrados);

} catch (err) {

console.error("Error al cargar contactos guardados:", err.message);

}

}

});

});

// Pone la función renderizarMisContactos disponible globalmente

window.renderizarMisContactos = renderizarMisContactos;