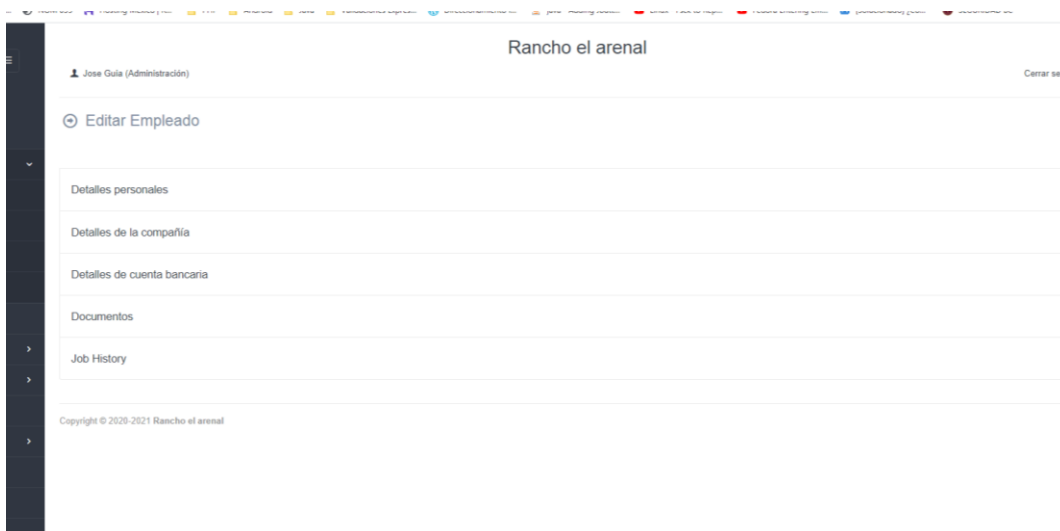


## Reporte del día sábado, 17 de abril de 2021

Se configuraron los controladores para mostrar la vista donde se encuentra el formulario para editar los datos de los empleados además de configurar para que los datos se obtengan mediante el id.

```
}  
}  
  
public function employee_edit($employee_code)  
{  
    $page_data['employee'] = $this->generic_model->selectEntity('user', array('user_code' => $employee_code), FALSE, NULL);  
    $page_data['page_name'] = 'employee_edit';  
    $page_data['user_code'] = $employee_code;  
    $page_data['page_title'] = get_phrase('edit_employee');  
    $this->load->view('backend/index', $page_data);  
}  
  
public function edit($employee_code)  
{  
    $employee = $this->employee_model->edit_employee($employee_code);  
    if ($employee == 'true') {  
        $this->session->set_flashdata('flash_message', get_phrase('data_updated_successfully'));  
        redirect(site_url('admin/employee/employee_edit/') . $employee_code, 'refresh');  
    }  
    redirect(site_url('admin/dashboard'), 'refresh');  
}  
  
public function personal_details($employee_code)  
{  
    // PERSONAL DETAILS view (employee_details)  
}
```

A continuación, se muestra la vista para editar los datos del empleado, se muestra por secciones una vez que se selecciona se mostrara el formulario correspondiente.



## Reporte del día sábado, 17 de abril de 2021

Se realizó la función para obtener los datos del empleado, una vez que se obtienen se guarda en una variable el dato que viene en la posición deseada, una vez guardada se agrega a el formulario para así mostrarlo.


```
117
118 <script>
119
120 _url="http://localhost/apistNominas/index.php/";
121
122 $(function () {
123 $.ajax({
124 url: _url+"employees/api/employees/id/",
125 type: 'POST',
126 dataType: 'JSON',
127 success: function(result) {
128
129 first_name = result.first_name;
130 last_name = result.last_name;
131 date_of_birth = result.date_of_birth;
132 gender = result.gender;
133 phone = result.phone;
134 local_address = result.local_address;
135 email = result.email;
136 nationality = result.nationality;
137 martial_status = result.martial_status;
138
139 $("#first_name").html(first_name);
140 $("#last_name").html(last_name);
141 $("#date_of_birth").html(date_of_birth);
142 $("#gender").html(gender);
143 $("#phone").html(phone);
144 $("#local_address").html(local_address);
145 $("#email").html(email);
146 $("#nationality").html(nationality);
147 $("#martial_status").html(martial_status);
148
149
```

A continuación, se muestra el formulario de para editar los datos personales del empleado, la función anterior muestra los datos correspondientes a cada campo que ya se han registrado anteriormente.

⊙ Editar Empleado


---

Detalles personales

Nombre	roberto guerra rodriguez
Apellidos	
Fecha de nacimiento	Mon, 07 Jun 1976
Género	Masculino
Teléfono	4181240552
Dirección local	santa cruz
Email	robertoguerra246@gmail.com
Nacionalidad	mexicana
Estado civil	Casado
Foto	

## Reporte del día sábado, 17 de abril de 2021

Se agregaron validaciones al formulario por ejemplo el formato correcto de un correo electronico y el limite de dígitos del numero telefonico entre otros.

Teléfono	<input type="text" value="4101240552"/>
Dirección local	<input type="text" value="santa cruz"/>
Email	<input type="text" value="robertoguerrero246@gmail.com"/>
Nacionalidad	<input type="text" value="mexicana"/>
Estado civil	<input type="text" value="Casado"/>
Foto	<div> <input type="button" value="Seleccionar imagen"/></div>
<input type="button" value="Guardar cambios"/>	

Se creo la funcion para poder actualizar los datos del empleado.

```
120
121 function editar(id) {
122   var form_data = new FormData(document.getElementById("personal_datos"));
123   $.ajax({
124     url: _url+"employees/api/employees/id/" +
125     id,
126     method: "POST",
127     headers: {
128       "X-API-KEY": "QroSystem0218",
129     },
130     data: form_data,
131     processData: false,
132     contentType: false,
133     cache: false,
134     async: false,
135     success: function (response) {
136       response = _response.message;
137       const Toast = Swal.mixin({
138         toast: true,
139         position: 'top-end',
140         showConfirmButton: false,
141         timer: 1500,
142         timerProgressBar: true,
143         didOpen: (toast) => {
144           toast.addEventListener('mouseenter', Swal.stopTimer)
145           toast.addEventListener('mouseleave', Swal.resumeTimer)
146         },
147       })
148       Toast.fire({
149         icon: 'success',
150         title: "Producto actualizado exitosamente",
151         onAfterClose: (toast) => {
152           window.location.reload();
153         }
154       })
155     },
156     error: function (error) {
157       _response = 1570; <stringify(error.responseText):
158     }
159   });
160 }
```