**1º Passo**

**Foi criado uma função ligada a um evento onclick para recuperar o valor dos campos que serão preenchidos.**

**function cadastrarDespesa(){**

**let ano = document.getElementById('ano')**

**let mes = document.getElementById('mes')**

**let dia = document.getElementById('dia')**

**let tipo = document.getElementById('tipo')**

**let descricao = document.getElementById('descricao')**

**let valor = document.getElementById('valor')**

**console.log(ano.value, mes.value, dia.value, tipo.value, descricao.value, valor.value)**

**}**

**2º Passo**

**Foi criado um objeto de classe que recebe como parâmetro os valores passados pela função**

**class Despesa {**

**constructor(ano, mes, dia, tipo, descricao, valor){**

**this.ano = ano**

**this.mes = mes**

**this.dia = dia**

**this.tipo = tipo**

**this.descricao = descricao**

**this.valor = valor**

**}**

**}**

**function cadastrarDespesa(){**

**let ano = document.getElementById('ano')**

**let mes = document.getElementById('mes')**

**let dia = document.getElementById('dia')**

**let tipo = document.getElementById('tipo')**

**let descricao = document.getElementById('descricao')**

**let valor = document.getElementById('valor')**

**let despesa = new Despesa(ano.value, mes.value, dia.value, tipo.value, descricao.value, valor.value)**

**console.log(despesa)**

**}**

**3º Passo**

**Foi criado uma função para gravar as informações do objeto em localStorage, e inserido a um start para está função dentro da função “cadastrarDespesa” já criada antes.**

**function cadastrarDespesa(){**

**let ano = document.getElementById('ano')**

**let mes = document.getElementById('mes')**

**let dia = document.getElementById('dia')**

**let tipo = document.getElementById('tipo')**

**let descricao = document.getElementById('descricao')**

**let valor = document.getElementById('valor')**

**let despesa = new Despesa(ano.value, mes.value, dia.value, tipo.value, descricao.value, valor.value)**

**gravar(despesa)**

**}**

**function gravar(d){**

**localStorage.setItem('despesa', JSON.stringify(d))**

**}**

**Como usamos o recurso setItem, toda vez que enviamos dados, eles sobrepõem um ao outro ao invés de adicionar mais um, devemos mudar isso com a inserção de uma lógica.**

**4º Iremos**

**Iremos implementar um objeto Bd que recebe a função gravar agora, e inserimos um índice dinamico**

**class Bd{**

**constructor() {**

**let id = localStorage.getItem('id')**

**if(id === null) {**

**localStorage.setItem('id', 0)**

**}**

**}**

**getProximoId() {**

**let proximoId = localStorage.getItem('id')**

**return parseInt(proximoId) + 1**

**}**

**gravar(d) {**

**let id = this.getProximoId()**

**localStorage.setItem(id, JSON.stringify(d))**

**localStorage.setItem('id', id)**

**}**

**}**

**let bd = new Bd()**

**Foi construído uma lógica de gravação onde se recebe uma lógica de indexação de id e um código onde se converte strings para JSON e após isso atualiza. Depois disso foi criado a lógica de indexação “getProximoId”**

**I – Retorna os índices de um determinado Array() ou os atribustos de um determinado objeto.**

**5º Iremos implementar uma lógica para validação das informações inseridas**

**Foi inserido dentro da Class “Despesa” uma função que contem uma lógica para identificar se dentre as informações enviadas pelo usuário contem algum campo undefined, null ou vazio.**

**class Despesa{**

**constructor(ano, mes, dia, tipo, descricao, valor){**

**this.ano = ano**

**this.mes = mes**

**this.dia = dia**

**this.tipo = dia**

**this.descricao = descricao**

**this.valor = valor**

**}**

**validarDados(){**

**for(let i in this){**

**if(this[i] == undefined || this[i] == '' || this[i] == null){**

**return false**

**}**

**}**

**return true**

**}**

**}**

**Dentro da função “cadastrarDespesas()” foi inserido um condicional para retornar informação para o usuário se a operação foi completa ou não, porem por enquanto está funcionando como debug.**

**function cadastrarDespesa() {**

**let ano = document.getElementById('ano')**

**let mes = document.getElementById('mes')**

**let dia = document.getElementById('dia')**

**let tipo = document.getElementById('tipo')**

**let descricao = document.getElementById('descricao')**

**let valor = document.getElementById('valor')**

**let despesa = new Despesa(ano.value, mes.value, dia.value, tipo.value, descricao.value, valor.value)**

**despesa.validarDados()**

**if(despesa.validarDados()){**

**//bd.gravar(despesa)**

**//Dialogo de Sucesso**

**console.log('Dados válidos')**

**}else{**

**//Dialogo de Erro**

**console.log('Dados invalidos')**

**}**

**}**

**5º Iremos implementar um aviso**

**Continuaremos o passo anterior que estava em fase de Debug de criar uma condicional, e nela faremos um incremento de inserir um aviso caso os dados estejam incompletos e caso os dados sejam armazenados com sucesso.**

**if(despesa.validarDados()){**

**bd.gravar(despesa)**

**$('#sucessoGravacao').modal('show')**

**}else{**

**//Dialogo de Erro**

**$('#erroGravacao').modal('show')**

**}**

**Esses Id que são chamados são Modais que foram inseridos no arquivo Html**

**//Formas de armazenamento de Dados em Browers**

**Storage**

**-Local Storage – Ficam armazenados no navegador**

**-Session Storage - Ficam armazenados no navegador enquanto o brower estiver ativo;**

**-IndexedDB**

**-Web SQL**

**-Cookies - funcionam como dados que são armazenados do lado do cliente, de modo que o servidor possa extrair e trabalhar com esses dados;**