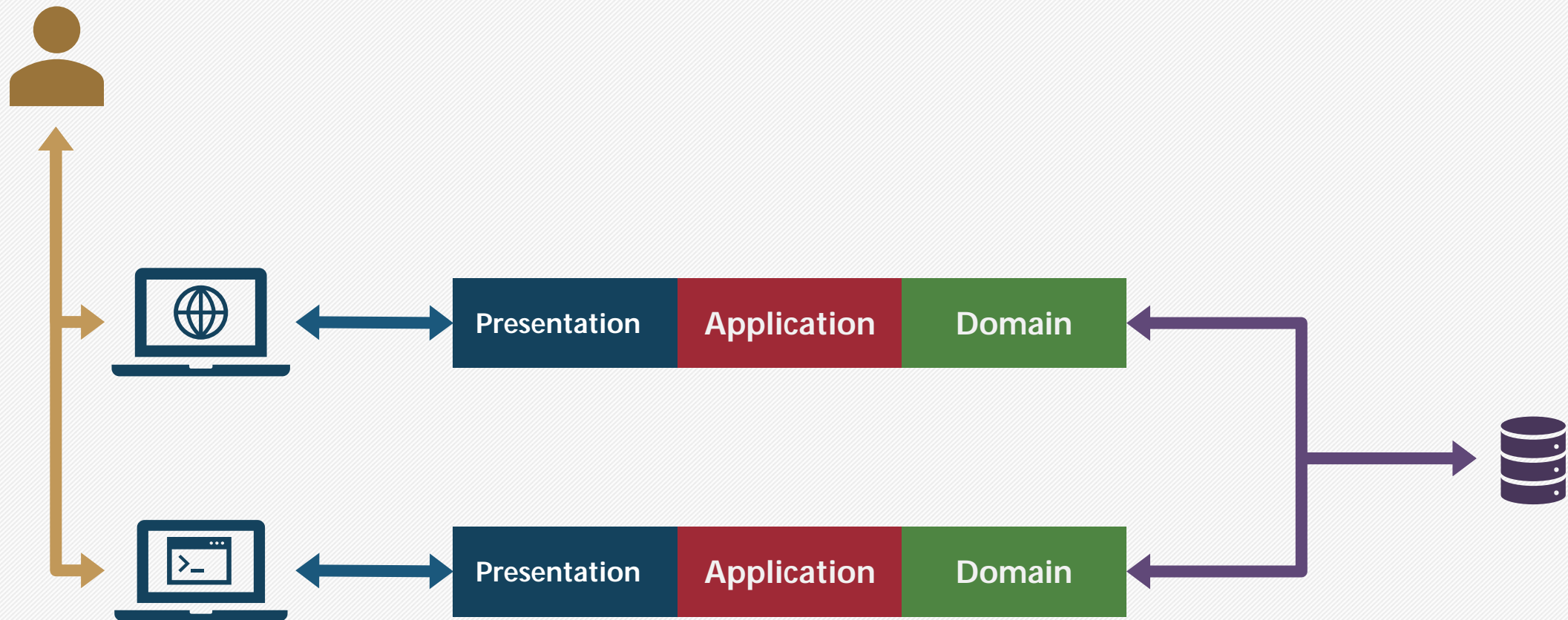
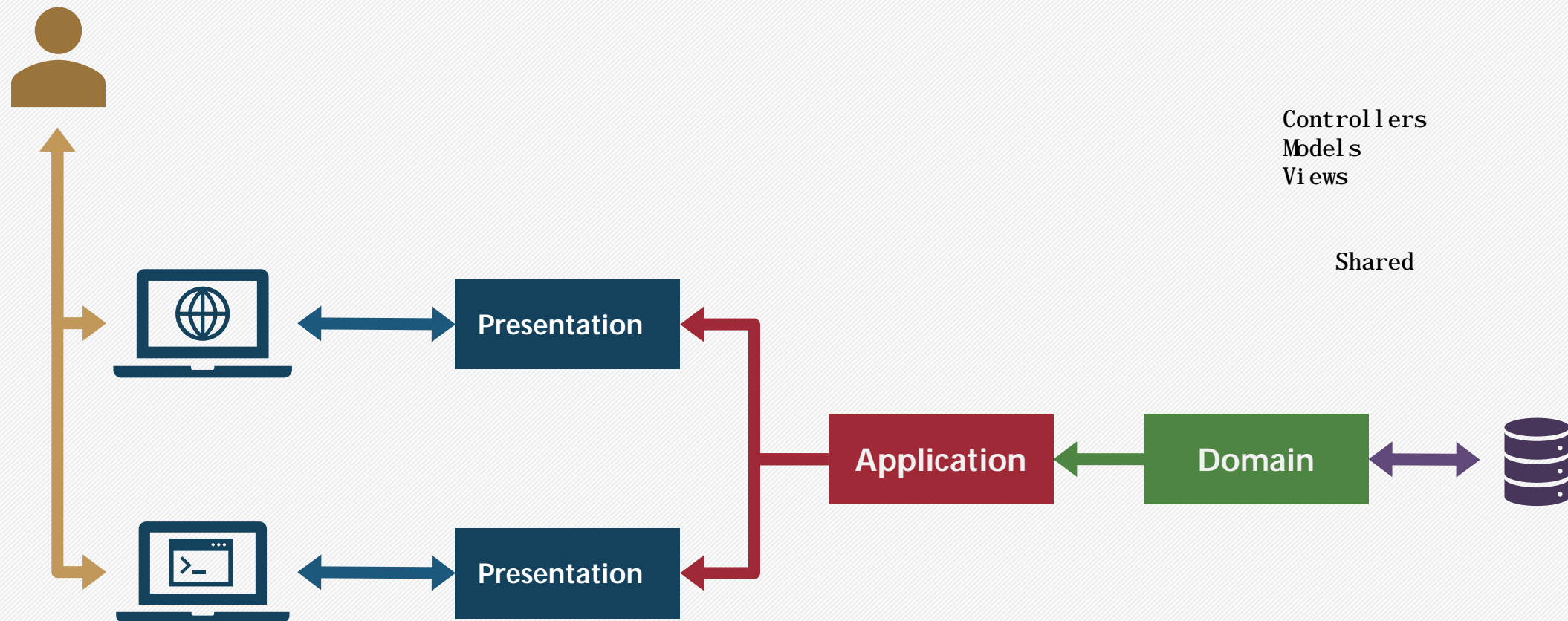


Problema



Solución



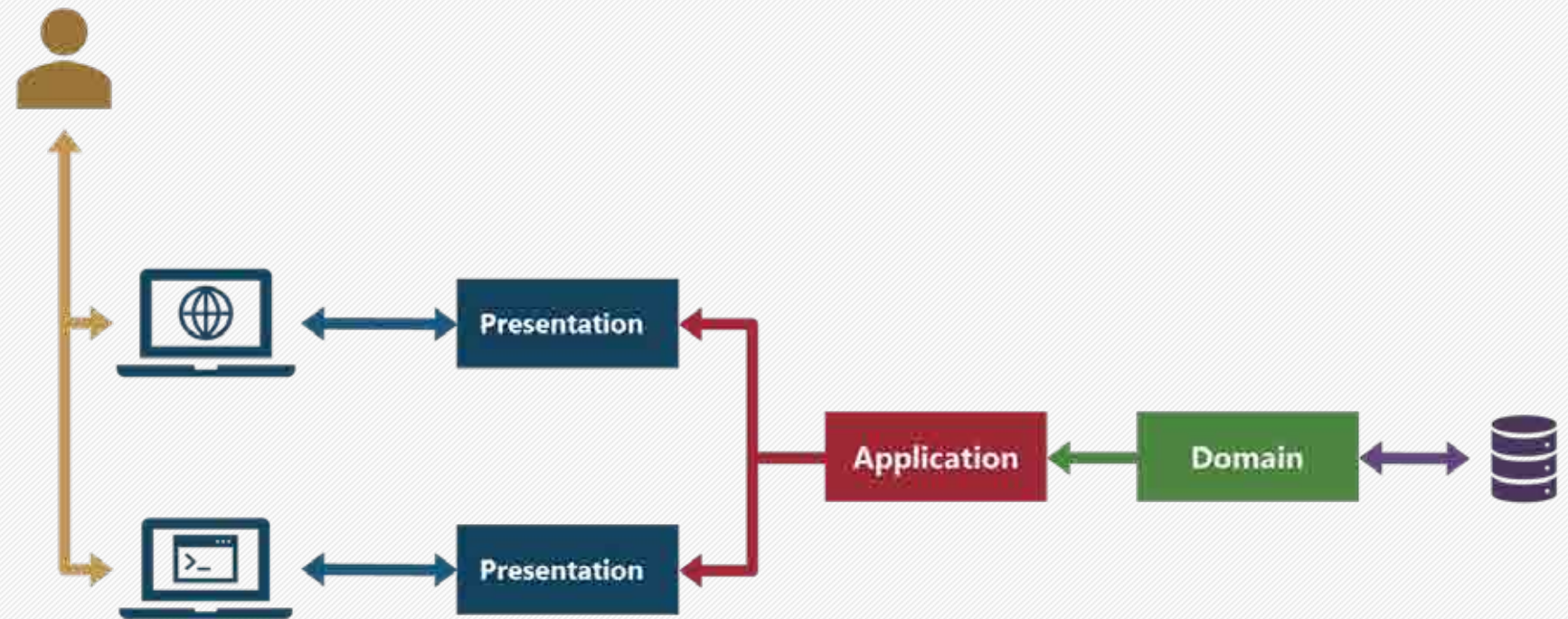
Conexión a base de datos en una aplicación de web mvc

<https://gist.github.com/jlcarrillog/c1f6a96269259a3bd40a37c8f7732a9a>

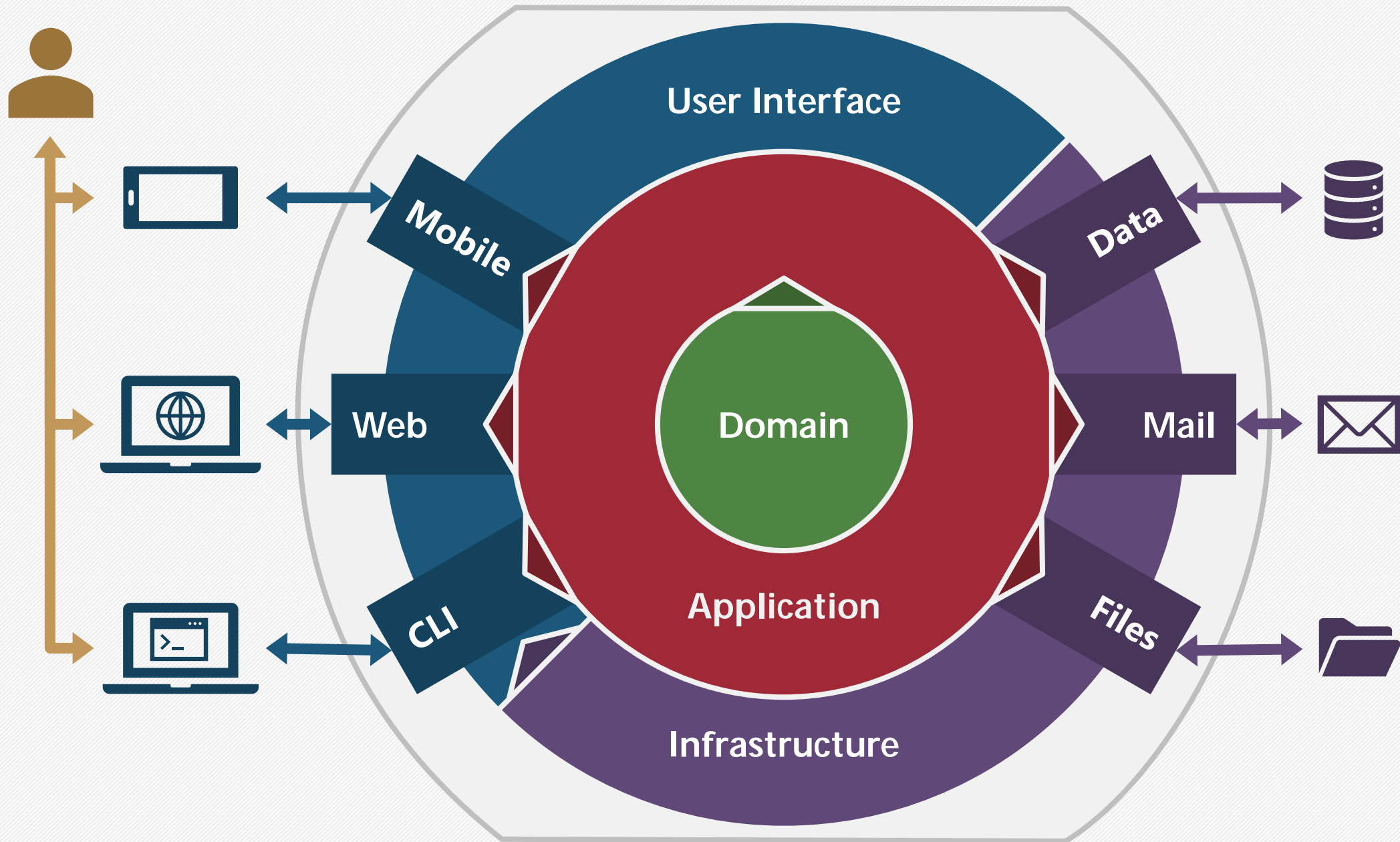
Entities

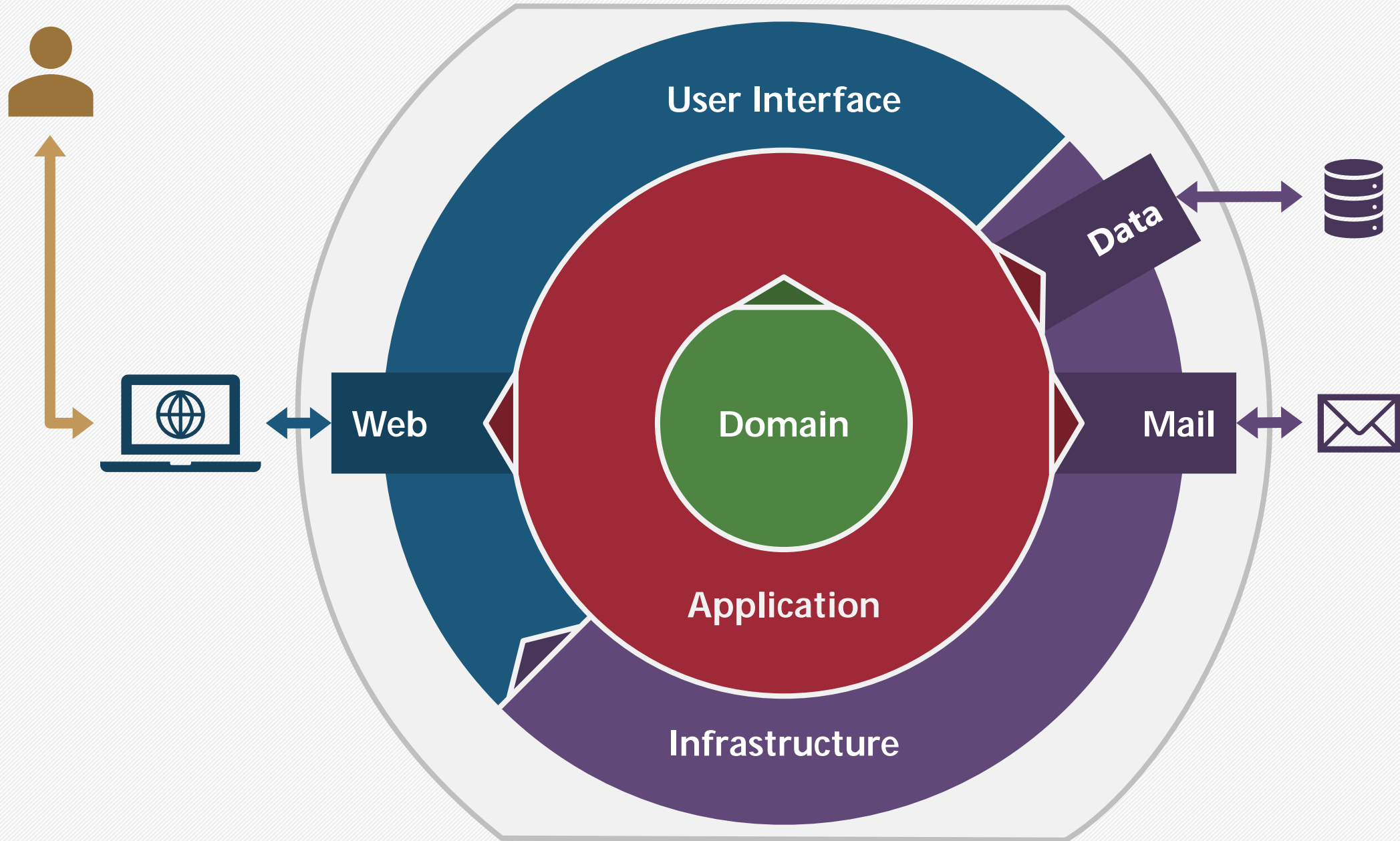
Controllers
Models
Views

Shared



Proyecto Arquitectura





Services

Entities

Enums

Notifications

Controllers

Models

Views

Shared

wwwroot

css

images

js

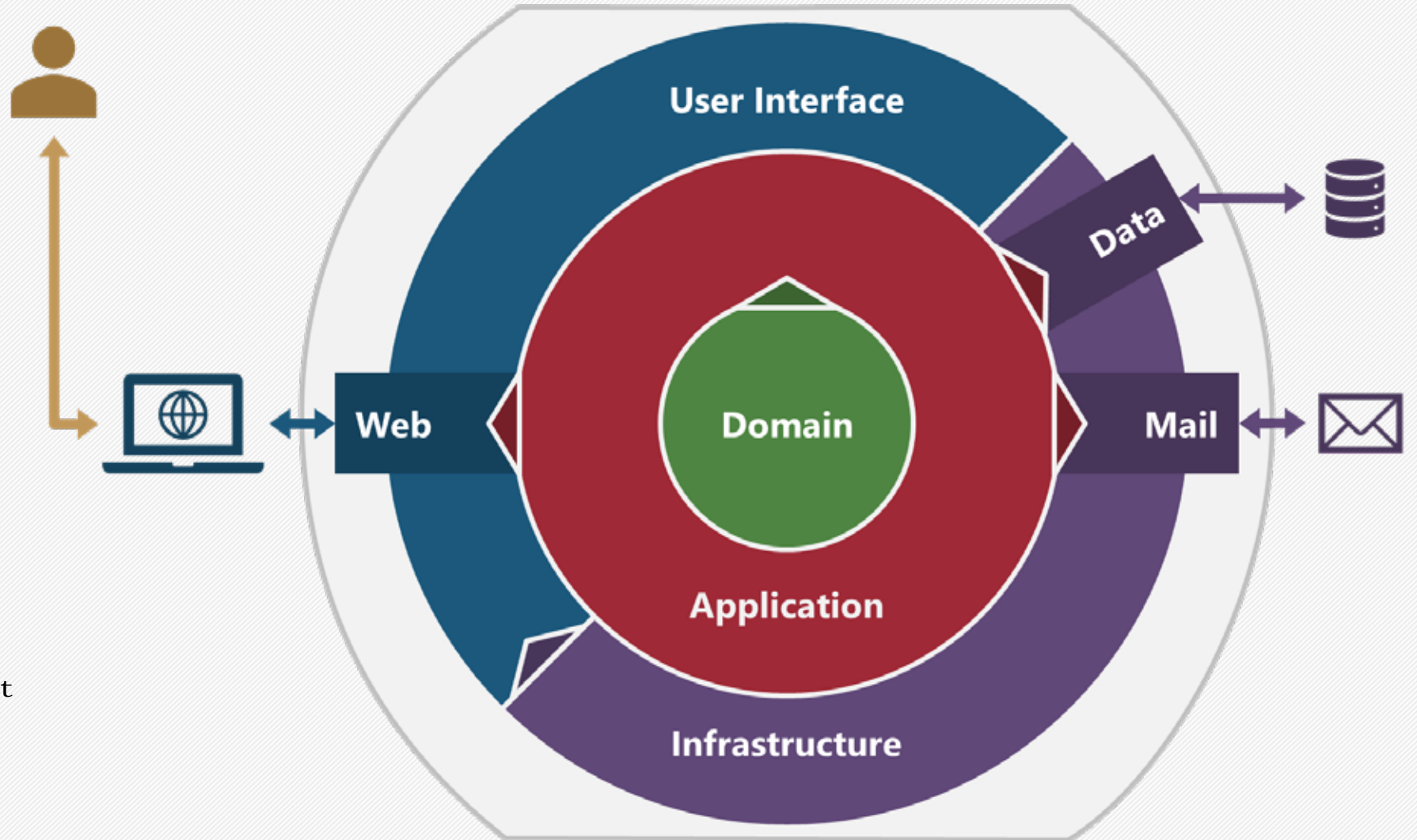
lib

Application. Test

Domain. Test

Infrastructure. Test

Presentation. WebApp. Test



Manejo de archivos

<https://gist.github.com/jlcarrillog/a075c61e691a4af74e659d8bd196ba75>

Manejo de archivos

Guardar en una **carpeta** del servidor y registrar el **path** en BD:

- Se envía y recibe el archivo en la acción
- Se obtiene el nombre del archivo y se concatena con la ruta
- Se guarda el archivo en la carpeta
- Se registra en la BD el path como texto

Guardar en un **repositorio** en un servicio en la nube y registrar el **path** en BD:

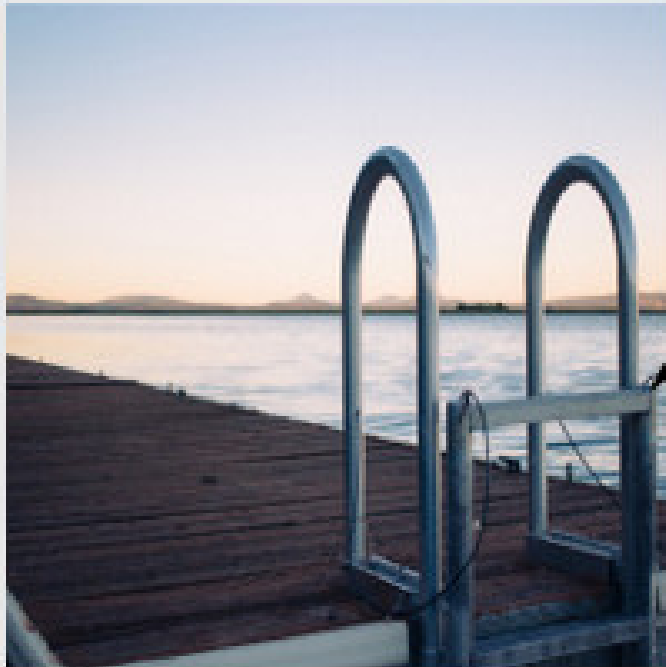
- Se envía y recibe el archivo en la acción (blobs)
- Se obtiene el nombre del archivo y se concatena con la ruta
- Se guarda el archivo en el servicio
- Se registra en la BD el path como texto

Guardar como **Base64** en BD como **texto**:

- Se envía y recibe el archivo en la acción
- Se convierte el archivo a Base64
- Se registra la conversión en la BD como texto

Sistema de numeración posicional que usa **64** como base.

Se usan el rango de caracteres **A-Z**, **a-z** y **0-9** en este orden para los primeros **62** dígitos, y símbolos escogidos para los **últimos dos varían**.



```
data:image/jpeg;base64,/9j/4AAQSkZJRgABAQEASABIAAD  
/4QCYRXhpZgAATU0AKgAAAAgABAEaAAUAAAABAAAAPgEbAAUAAAABAA  
AARgEoAAMAAABAAIAAIIdpAAQAAAABAAAATgAAAAAAABIAAAAAQAAA  
EgAAAABAAWQAAAHAAABBDAYMTCgAAAHAAABBDAXMDCgAQADAAAAAQAB  
AACgAgAEAAAAAQAAASygAwAEAAAAAQAAAMg .....
```

```
<form asp-action="Insertar" enctype="multipart/form-data">
    Foto: <input type="file" name="file" accept=".jpg, .png, .jpeg, .gif" required><br />
    <button>Insertar</button>
</form>
```

```
public class HomeController : Controller
{
    public IActionResult Index()
    {
        return View();
    }

    [HttpPost]
    public IActionResult Index(
        IFormFile file)
    {

        return View();
    }
}
```

```
@model string
Foto: 
```


Envió de email

<https://gist.github.com/jlcarrillog/6983702ee1b217b6736d2bea00a54034>

```
<form asp-action="Create" enctype="multipart/form-data">
  Nombre: <input name="nombre" required />
  Correo: <input name="correo" required />
  <button>Crear</button>
</form>
```

```
public class HomeController : Controller
{
    private readonly SmtpClientEmailService _email;

    public HomeController(IConfiguration configuration)
    {
        var config = configuration.GetSection("Smtp");
        _email = new SmtpClientEmailService(config["Displayname"], config["Address"], config["Host"],
                                            int.Parse(config["Port"]), config["Username"], config["Password"]);
    }

    public IActionResult Index()
    {
        return View();
    }

    [HttpPost]
    public async Task<IActionResult> Index(string correo, string nombre)
    {
        await _email.SendEmailAsync(correo, "Asunto", $"<h4>Hola {nombre}</h4>", true);
        return View();
    }
}
```

```
{
  "Smtp": {
    "Host": "smtp.office365.com",
    "Port": 587,
    "Username": "***@mail.com",
    "Password": "***",
    "Address": "***@mail.com",
    "Displayname": "***"
  }
}
```



```

using System;
using System.Net;
using System.Net.Mail;
using System.Threading.Tasks;

namespace IM04a2.Controllers
{
    public class SmtplibClientEmailService
    {
        private SmtplibClient client { get; set; }
        private MailMessage mail { get; set; }

        public SmtplibClientEmailService(string fromName, string fromAddress, string host, int port, string userName, string password)
        {
            client = new SmtplibClient();
            client.Host = host;
            client.Port = port;
            client.UseDefaultCredentials = false;
            client.DeliveryMethod = SmtplibDeliveryMethod.Network;
            client.Credentials = new NetworkCredential(userName, password);
            client.TargetName = "STARTTLS/smtplib.office365.com";
            client.EnableSsl = true;
            mail = new MailMessage();
            mail.From = new MailAddress(fromAddress, fromName);
        }

        public async Task SendEmailAsync(string email, string subject, string message, bool isBodyHtml)
        {
            mail.Subject = subject;
            mail.IsBodyHtml = isBodyHtml;
            mail.Body = message;
            mail.BodyEncoding = System.Text.Encoding.UTF8;
            mail.SubjectEncoding = System.Text.Encoding.UTF8;
            mail.To.Add(email);
            try
            {
                client.Send(mail);
                await Task.FromResult(0);
            }
            catch (Exception)
            {
                throw;
            }
        }
    }
}

```