

“Año del Bicentenario, de la consolidación de nuestra Independencia, y de la conmemoración de las heroicas batallas de Junín y Ayacucho”



Facultad de ingeniería

GRUPO 1

SEMANA 06

Taller de Programación Web

Docente: Oscar Alberto BARNETT CONTRERAS

Integrantes:

Cesar del Aguila U17392165

Alonso Giraldo Martinez U23232870

Maykol Cristian, Cahuana Tambine U20306285

INDICE

Tabla de contenido

RESUMEN EJECUTIVO.....	3
INTRODUCCION	3
CONTEXTO DE APLICACIÓN	3
IDENTIFICACIÓN Y DESCRIPCIÓN DEL PROBLEMA	4
DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN	4
CRONOGRAMA DE ACTIVIDADES.....	4
Estado del Arte	6
Regionalmente:.....	7
Nacionalmente:	7
Internacionalmente:	8
TECNOLOGÍA UTILIZADA.....	8
Front-End: HTML-CSS-JSS.....	8
Framework Front-End: Bootstrap, etc.	9
Lenguaje de Programación Back-End: PHP.	9
Servidor Web: Apache de XAMPP.....	9
Motor de Base de Datos: MySQL.....	9
Control de versiones: Git / GitHub (Opcional).....	10
DESCRIPCIÓN DEL SISTEMA	10
MODELO:	19
CONTROLADOR:.....	22
VISTA:	23
GLOSARIO:	27
Diagramas varios (que se han utilizado en el desarrollo del proyecto):	29
1. Esquema del Sitio Web.....	29
2. Diagrama caso de uso.....	30
3. Modelo entidad relación de la BD	31
4. Sketch / Prototipos o similares.	32

RESUMEN EJECUTIVO

En la actualidad el sector privado de la educación (colegios particulares) ha tenido un incremento significativo con respecto a la pública siendo que la educación básica regular alberga alrededor de ocho millones de alumnos en el país. De ellos, un 30% asiste a escuelas privadas. En este panorama, las instituciones privadas de educación requieren de una web que les permita centralizar y facilitar la atención y poner a disposición herramientas que tanto alumnos como maestros puedan usar. Asimismo, este sistema requiere de una serie de funcionalidades que, dependiendo del tamaño de la institución, puede ir desde un pequeño servidor local que aloje toda la información, hasta un super servidor en la nube que cuente con redundancia, respaldos y alta disponibilidad para garantizar la continuidad de sus servicios. En respuesta a esa necesidad, nace este proyecto que propone suministrar una web en donde tanto profesores, alumnos y personal administrativo pueda tener rápido acceso a diversos recursos y funciones, para poder cumplir su rol para con la institución.

INTRODUCCION

En este trabajo pretendemos reflejar nuestro aprendizaje a lo largo del curso, comprendiendo las diferentes herramientas instruidas en clase para poder desarrollar la web antes ya mencionada. Asimismo, debido a la alta demanda de portales web que existen a lo largo del país, se tomaron como referencia algunas funcionalidades de otros portales ya existentes. Finalmente, todo el desarrollo será publicado en un servidor Web para lo cual se contará con un dominio público bajo el nombre xyz.com.pe

CONTEXTO DE APLICACIÓN

Debido a los avances tecnológicos actuales, es posible que todo el desarrollo será en la nube, haciendo que la edición de esta sea más flexible y garantizando la seguridad de la información que se maneje en ese entorno. Para ello, hemos creado un contenedor GitHub en donde estamos creando toda la estructura de la Web, para luego pasar a publicarla por medio de un dominio en punto.pe, cuyo nombre final se definirá internamente.

La aplicación será desarrollada en HTML, posteriormente se le ira integrando diseño basado en CSS, para posteriormente insertar funcionalidades propias de JavaScript y tener una solución completa y funcional para el uso del usuario final. La solución pretende simular una intranet y para ello tendrá que contar una Base de Datos MySQL que será considerada dentro del desarrollo en la tercera etapa.

IDENTIFICACIÓN Y DESCRIPCIÓN DEL PROBLEMA

La institución educativa XYZ, no cuenta con una página web y menos con una intranet, por ello todas las gestiones siguen siendo en papel, lo que hace que la información que se maneje no tenga ningún factor de seguridad, lo cual no lo hace cuestionable. Así mismo, al no contar con una intranet, la información no es confiable.

Adicional a ello, en el mundo en que vivimos es esencial para el crecimiento de cualquier empresa contar con una imagen digital que pueda cumplir la función de promover la marca a través de marketing digital y redes sociales.

DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN

Esta etapa del proyecto se diseñará al finalizar el trabajo adjuntando el manual de usuario en donde se detallará el nivel funcional de la web

CRONOGRAMA DE ACTIVIDADES

Para el avance continuo del proyecto se definió reuniones semanales todos los domingos a las 8pm, en donde se definirán las actividades a realizar por cada miembro del equipo, y los viernes en la tarde se validará y corregirá todo el avance en general. Los días de presentación de avance, se reunirán todos los miembros del equipo 1 hora antes de la clase para repasar la presentación y definir roles. Presentador 1,2,3 y 4 respectivamente.

[illegible]

Estado del Arte

MOTIVACION

En cualquier proceso de investigación es necesario seguir diversos pasos, todos ellos fundamentales, para abordar cualquier problema. Por ellos queremos tomar como referencia los cambios tecnológicos que se vienen realizando en el ámbito educativo de nuestro país. Así mismo, recalcar las exigencias de la sociedad globalizada, de estos factores, nace la motivación para realizar nuestro trabajo desarrollo de una página web para un colegio particular, ya que pudimos observar la creciente demanda de entornos digitales, y como estos influyen dentro del funcionamiento y futuro de todas las instituciones hoy en día.

DIAGNOSTICO

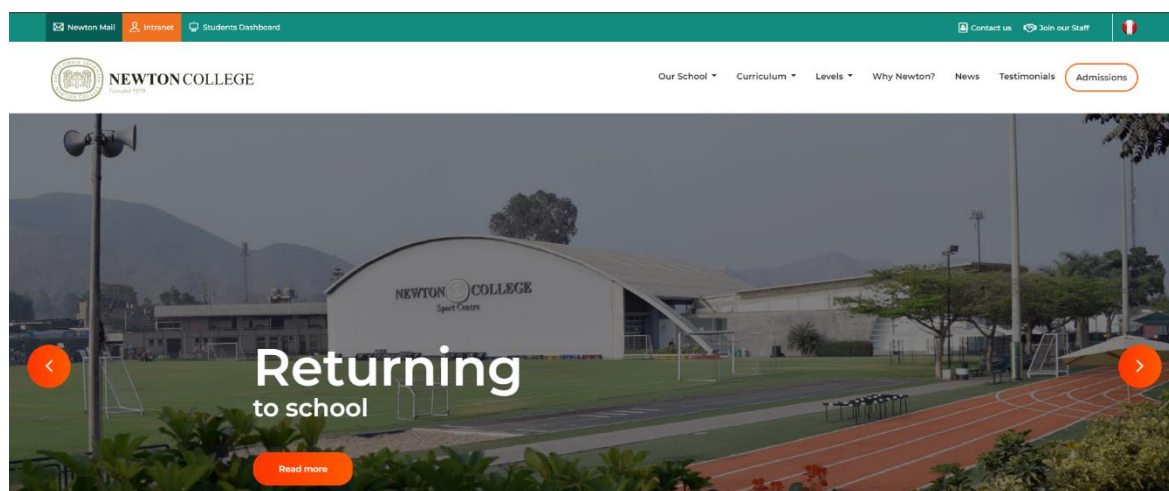
Podemos observar por el FODA inicial, que, dentro de las fortalezas de la institución, se encuentra la ubicación privilegiada y la alta demanda de colegios en la zona. De igual manera, una de las debilidades más notables, es la falta de una página Web en donde puedan interactuar tanto alumnos como docentes, así como público en general.

ANALISIS FODA

FORTALEZAS	DEBILIDADES
Buen clima laboral	Falta de apoyo familiar
Liderazgo de parte de los directivos	Falta de áreas verdes para instalar un huerto escolar
Seguimiento y apoyo académico	Estudiantes con pocos valores morales influyen en el resto
Compromiso del personal docente	evaluación sumativa y no formativa de algunos docentes
Formación continua del personal docente	Poco uso de las Tics
Todo el personal docente cuenta con maestría	Falta de apoyo de los padres en las actividades extracurriculares
Escuela para padres	
Actividades extracurriculares	
Infraestructura y laboratorios informáticos	
OPORTUNIDADES	AMENAZAS
Docentes actualizados en la propuesta pedagógica	Falta de tiempo de padres de familia
Proyectar la escuela hacia la comunidad	Conflictos familiares que afecten el rendimiento académico
capacitación y actualización docente constante	Falta de recursos económicos
Motivar a alumnos a participar en actividades extracurriculares	Falta de interés de los alumnos y/o padres
Seguimiento de alumnos con problemas de aprendizaje	Influencia de actividades (videojuegos, tv, celulares,etc)

Habiendo analizado el FODA de la institución, podemos decir que nuestra referencia para mejorar y a donde queremos son las siguientes:

Regionalmente: El colegio emblema y el cual cuenta con varios premios, así como una pagina Web muy completa y dinámica, el Newton College (<https://www.newton.edu.pe/>). De esta institución podemos resaltar su educación personalizada enfatizando y explotando las habilidades de cada estudiante de manera eficiente, logrando resultados mas que satisfactorios tanto para los docentes, como para los padres. Todo esto se ve claramente plasmado en la web que tienen en línea.



Nacionalmente: Escalando un poco en el grado académico, podemos tomar a la PUCP como un referente debido a su diversa gama de especializadas, tanto en prepagado como en postgrado. Asimismo, es una de las universidades que cuenta con un colegio el cual te prepara directamente para el la educación dentro de la Universidad del mismo grupo. (<https://www.pucp.edu.pe/>)



Internacionalmente: Sin duda el colegio que tomamos como referencia es el colegio Montessori en Colombia. Este aparte de haber sido elegido como el mejor colegio de Sudamérica cuenta con una pagina web digna del prestigio que representa la institución. Apartados interesantes como el de Boletines, Eventos, Noticias, etc. Nos demuestran que una buena página web, transmite mucho lo que hay de trasfondo. (<https://montessori.edu.co/es/>)



TECNOLOGÍA UTILIZADA

En el desarrollo de este proyecto tecnológico se ha utilizado un conjunto de herramientas tanto para el front-end como para el back-end, con el objetivo de crear una experiencia web dinámica y atractiva para los usuarios.

Front-End: HTML-CSS-JSS

HTML (HyperText Markup Language): Es el lenguaje de marcado estándar para crear páginas web. Se utiliza para estructurar el contenido de la página y tiene una versión HTML5 que es ampliamente adoptada, ofreciendo nuevas características semánticas y de multimedia. Es compatible con la mayoría de los navegadores modernos, aunque se requieren polyfills (polyfills) para la compatibilidad con versiones anteriores de Internet Explorer. Alternativas como XHTML ofrecen una sintaxis más estricta.

CSS (Cascading Style Sheets): Es un lenguaje utilizado para describir la presentación de un documento HTML. Controla el diseño, formato y estilo de los elementos HTML. CSS3 proporciona numerosas funcionalidades avanzadas como transiciones, animaciones y flexbox para diseño responsivo. La compatibilidad con los navegadores varía, y se pueden requerir prefijos de proveedores para algunas

propiedades. Otras alternativas incluyen preprocesadores como Sass o Less, que añaden características como variables y mixins.

JavaScript: Es un lenguaje de programación que se utiliza para agregar interactividad y funcionalidad a las páginas web. Puede utilizarse para crear efectos dinámicos, validar formularios, entre otras cosas. ECMAScript 6 (también conocido como ES6 o ES2015) introduce nuevas características como clases, arrow functions, y destructuring. Aunque es ampliamente compatible con los navegadores modernos, se necesita transpilación para admitir versiones anteriores. Frameworks como React, Angular y Vue.js ofrecen abstracciones y herramientas para facilitar el desarrollo.

Framework Front-End: Bootstrap, etc.

Bootstrap: Es un framework de desarrollo front-end que facilita la creación de interfaces web responsivas y estéticamente atractivas. Proporciona una amplia variedad de componentes y estilos predefinidos que permiten diseñar rápidamente interfaces de usuario modernas y adaptables a diferentes dispositivos y tamaños de pantalla.

Lenguaje de Programación Back-End: PHP.

PHP: Es un lenguaje de programación de uso general especialmente adecuado para el desarrollo web. Se utiliza para crear aplicaciones web dinámicas y sitios web interactivos, procesar formularios, acceder a bases de datos y realizar otras tareas del lado del servidor.

Servidor Web: Apache de XAMPP.

Apache de XAMPP: Apache es un servidor web de código abierto ampliamente utilizado que proporciona un entorno robusto y confiable para alojar sitios web y aplicaciones web. XAMPP es un paquete de software que incluye Apache, MySQL, PHP y otros componentes necesarios para el desarrollo web local.

Motor de Base de Datos: MySQL.

MySQL: Es un sistema de gestión de bases de datos relacional de código abierto ampliamente utilizado. Se utiliza para almacenar y gestionar datos de forma eficiente, permitiendo el acceso rápido y seguro a la información almacenada en la aplicación web.

Control de versiones: Git / GitHub (Opcional)

Git / GitHub: Git es un sistema de control de versiones distribuido que permite realizar un seguimiento de los cambios en el código fuente durante el desarrollo de software. GitHub es una plataforma en línea que permite alojar proyectos de Git, colaborar con otros desarrolladores, realizar seguimiento de problemas y realizar despliegues de código de forma colaborativa. Su uso es opcional pero altamente recomendado para el desarrollo colaborativo y la gestión eficiente del código fuente.

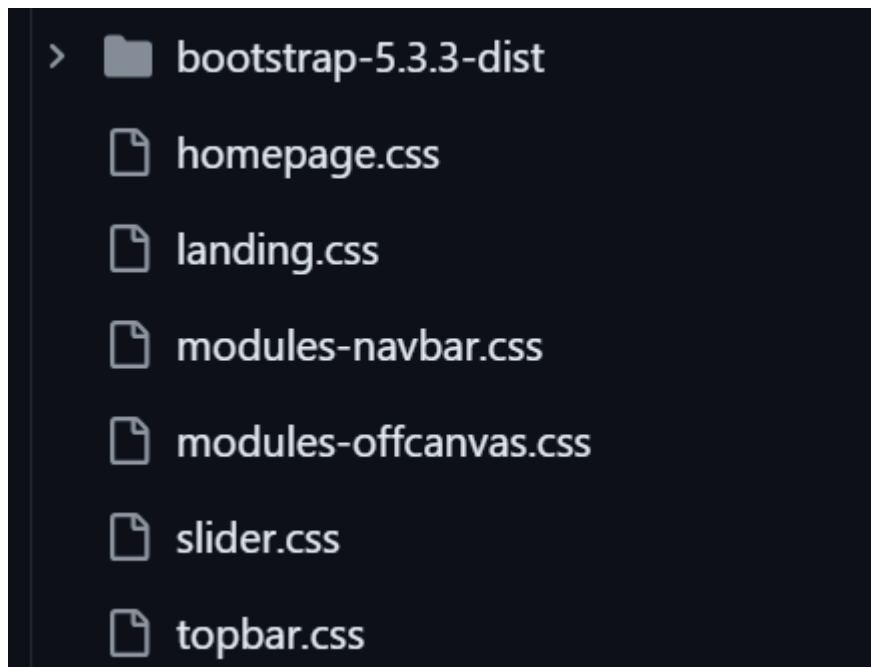
DESCRIPCIÓN DEL SISTEMA

Nuestra página web está orientada a tener una comunicación rápida y fluida con la información a la cual se busca obtener, además, que sea atractiva visualmente para el usuario. Por ello se planteó el desarrollo de la siguiente forma:

Index: Será la página predeterminada de nuestro sitio web:

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="utf-8" />
6   <meta name="viewport" content="width=device-width, initial-scale=1" />
7   <title>Colegio XYZ</title>
8   <script src="https://kit.fontawesome.com/c4f86b63aa.js" crossorigin="anonymous"></script>
9   <link href="/styles/bootstrap-5.3.3-dist/css/bootstrap.min.css" rel="stylesheet" />
10  <link rel="preconnect" href="https://fonts.googleapis.com">
11  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
12  <link href="https://fonts.googleapis.com/css2?family=Schibsted+Grotesk:ital,wght@0,400..900;1,400..900&display=swa
13    rel="stylesheet">
14  <link href="/styles/homepage.css" rel="stylesheet" />
15  <link href="/styles/slider.css" rel="stylesheet" />
16 </head>
17
18 <body>
19   <nav class="navbar navbar-expand-lg bg-body-tertiary">
20     <div class="container-fluid">
21       <a class="navbar-brand" href="#">Colegio XYZ</a>
22       <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarNav"
23         aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">
24         <span class="navbar-toggler-icon"></span>
25       </button>
26       <div class="collapse navbar-collapse" id="navbarNav">
27         <ul class="navbar-nav">
```

Styles: para realizarlos estamos usando Bootstrap, que es un framework gratuito de código abierto utilizado para facilitar la creación de páginas web, en nuestro caso estamos usando los siguientes:



Ejemplo del modules-navbar.css:

```
1  body {
2    padding-top: 4.5rem;
3  }
4
5  .navbar-custom {
6    background: #ffffff; /*linear-gradient(to bottom right, #fc5447, #fe9187);*/
7    box-shadow: none;
8    border-bottom: #fc5447;
9    color: #fc5447;
10   box-shadow: 0 0px 5px 0px rgba(0,0,0,.2);
11   padding-top: 5px;
12   padding-bottom: 5px;
13 }
14
15 .email.row{
16   font-size: 75%;
17 }
18
19 .custom-toggler .navbar-toggler-icon {
20   background-image: url("data:image/svg+xml;charset=utf8,%3Csvg viewBox='0 0 32 32' xmlns='http://www.w3.org/2000/svg'");
21 }
22
23 .custom-toggler.navbar-toggler {
24   border-color: #fc5447;
25 }
26
```

Ejemplo del modules-offcanvas.css:

```
1  .offcanvas-body a {
2      font-weight: bold;
3      padding-left: 10px;
4      border-radius: 5px;
5      font-size: 110%;
6      padding: 5px 5px 5px 5px;
7      margin: 0px 0px 10px 0px;
8      font-family: "Open Sans", "sans-serif";
9
10 }
11
12 .offcanvas-header {
13     font-weight: bold;
14     padding: 11px 16px;
15     background-color: #fc5447;
16     font-family: "Open Sans";
17     color: #ffffff !important;
18 }
19
20 .offcanvas-body {
21     background-color: #ffffff;
22 }
23
```

Ejemplo del topbar.css:

```
1  body {
2    font-family: "Trebuchet MS", sans-serif;
3    margin: 0;
4    padding: 0;
5  }
6
7  header {
8    background: linear-gradient(to bottom right, #fc5447, #fe9187);
9    display: flex;
10   color: #ffffff;
11   align-items: center;
12 }
13
14 .logo {
15   display: inline-block;
16   padding: 5px 0px 5px 13px;
17 }
18
19 .btn-top {
20   display: flex;
21   align-items: center;
22   padding-left: 20px;
23 }
24
25 .btn-top button {
26   display: flex;
27   align-items: center;
28   background: none;
```

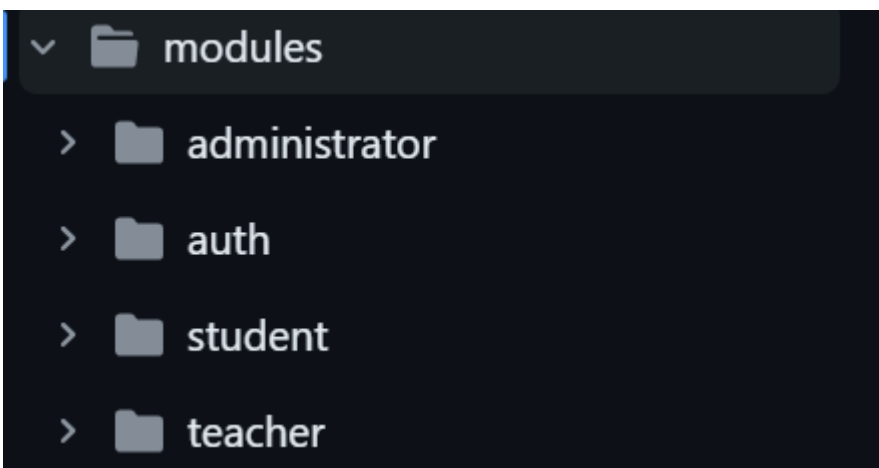
Scripts: es un código que nos va a permitir realizar distintas funcionalidades, para nuestro caso estamos utilizando un script para dar un efecto visual en la página, más concretamente llamado SlideShow, ejemplo del código:

```

1   let slideIndex = 1;
2   showSlides(slideIndex);
3
4   // Next/previous controls
5   function plusSlides(n) {
6       showSlides(slideIndex += n);
7   }
8
9   // Thumbnail image controls
10  function currentSlide(n) {
11      showSlides(slideIndex = n);
12  }
13
14  function showSlides(n) {
15      let i;
16      let slides = document.getElementsByClassName("mySlides");
17      let dots = document.getElementsByClassName("dot");
18      if (n > slides.length) {slideIndex = 1}
19      if (n < 1) {slideIndex = slides.length}
20      for (i = 0; i < slides.length; i++) {
21          slides[i].style.display = "none";
22      }
23      for (i = 0; i < dots.length; i++) {
24          dots[i].className = dots[i].className.replace(" active", "");
25      }
26      slides[slideIndex-1].style.display = "block";
27      dots[slideIndex-1].className += " active";

```

Módulos: Actualmente estamos trabajando con 4 módulos los cuáles son los siguientes:



Cada uno de estos este trabajado en código HTML, PHP y maneja su propio style en CSS:

administrador HTML:

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5     <meta charset="utf-8" />
6     <meta name="viewport" content="width=device-width, initial-scale=1" />
7     <title>Panel Admin</title>
8     <script src="https://kit.fontawesome.com/c4f86b63aa.js" crossorigin="anonymous"></script>
9     <link href="/styles/bootstrap-5.3.3-dist/css/bootstrap.min.css" rel="stylesheet" />
10    <link href="https://fonts.googleapis.com/css?family=Open+Sans" rel="stylesheet" />
11
12    <link rel="stylesheet" type="text/css" href="/styles/modules-navbar.css" />
13    <link rel="stylesheet" type="text/css" href="/styles/modules-offcanvas.css" />
14    <link href="/modules/administrator/styles.css" rel="stylesheet" />
15 </head>
16
17 <body>
18     <nav class="navbar navbar-custom fixed-top border-bottom border-body navbar-dark custom-toggler"
19         aria-label="Dark offcanvas navbar">
20         <div class="container-fluid">
21             <div>
22                 <div>
23                     <div class="col-12 d-flex justify-content-end">
24                         <div class="container text-center">
25                             <div class="row">
26                                 <div class="col container text-middle navbar-data-text">
```

Style CSS:

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5     <meta charset="utf-8" />
6     <meta name="viewport" content="width=device-width, initial-scale=1" />
7     <title>Panel Admin</title>
8     <script src="https://kit.fontawesome.com/c4f86b63aa.js" crossorigin="anonymous"></script>
9     <link href="/styles/bootstrap-5.3.3-dist/css/bootstrap.min.css" rel="stylesheet" />
10    <link href="https://fonts.googleapis.com/css?family=Open+Sans" rel="stylesheet" />
11
12    <link rel="stylesheet" type="text/css" href="/styles/modules-navbar.css" />
13    <link rel="stylesheet" type="text/css" href="/styles/modules-offcanvas.css" />
14    <link href="/modules/administrator/styles.css" rel="stylesheet" />
15 </head>
16
17 <body>
18     <nav class="navbar navbar-custom fixed-top border-bottom border-body navbar-dark custom-toggler"
19         aria-label="Dark offcanvas navbar">
20         <div class="container-fluid">
21             <div>
22                 <div>
23                     <div class="col-12 d-flex justify-content-end">
24                         <div class="container text-center">
25                             <div class="row">
26                                 <div class="col container text-middle navbar-data-text">
```

Student:

```
2  <!DOCTYPE html>
3  <html lang="en">
4    <head>
5      <meta charset="UTF-8" />
6      <meta name="viewport" content="width=device-width, initial-scale=1.0" />
7      <title>Document</title>
8      <link rel="stylesheet" type="text/css" href="../../styles/topbar.css" />
9    </head>
10   <body>
11     <header>
12       <div class="logo">
13         
19       </div>
20       <div class="btn-top">
21         <button>Perfil</button>
22         <button>Cursos</button>
23         <button>Horarios</button>
24         <button>Profesores</button>
25       </div>
26
```

Teacher:

```

1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Portal del Profesor - Colegio XYZ</title>
7   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
8   <link href="https://cdn.jsdelivr.net/npm/@fortawesome/fontawesome-free@6.0.0-beta3/css/all.min.css" rel="stylesheet">
9   <style>
10     body {
11       font-family: Arial, sans-serif;
12       margin: 0;
13       padding: 0;
14     }
15     .header-container {
16       display: flex;
17       align-items: center;
18       justify-content: space-between;
19       background-color: #f9f9f9;
20       padding: 10px;
21     }
22     .header-container h1 {
23       margin: 0;
24       color: black;
25       text-align: center;
26       order: 2;
27       flex: 1;

```

Landing page:

```

1 <!DOCTYPE html>
2 <html>
3
4 <head>
5   <meta charset="UTF-8">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Campos de Entrada</title>
8 </head>
9
10 <body style="
11   margin: 0;
12   display: flex;
13   justify-content: center;
14   align-items: center;
15   height: 100vh;
16   background-color: #fff3f1;
17   font-family: 'Trebuchet MS', sans-serif;
18   padding: 0;
19 ">
20   <div class="central-container" style="
21     background-color: rgb(255, 255, 255);
22     border: 3px solid;
23     border-color: #fc5447;
24     border-radius: 15px;
25     box-shadow: 0 0 20px rgba(0, 0, 0, 0.2);
26   ">
27     <div class="element-container" style="padding: 30px 70px; text-align: center;">
28       <p style="font-size: 140%; padding: 0px 30px 10px 30px; margin: 0px;"><h>INICIO DE SESIÓN</h></p>

```

Arquitectura MVC:

Esta arquitectura fue implementada en nuestros módulos de alumno y profesor para tener un mejor mantenimiento del código utilizado, además que cada componente tiene una responsabilidad definida dentro de nuestro entorno y se puede modificar de forma independiente a los demás, de igual forma el uso de las condicionales y las estructuras repetitivas están presentes tanto para nuestro controlador como también para las vistas.

MODELO:

- En nuestro modelo estamos utilizando los siguientes arreglos:
 - BuscarClasePorDia(\$dia): Este método busca y devuelve todas las clases programadas para un día específico pasado como argumento. Utiliza un bucle foreach para iterar sobre las clases y un if dentro del bucle para verificar si el día está presente en la descripción de la clase (\$clase[1]).
- -ObtenerPromediosPorMateria(): Este método calcula el promedio de calificaciones por materia utilizando los datos simulados de notas recientes. Primero, recorre el arreglo de notas utilizando un bucle foreach. Luego, utiliza una estructura condicional (if) para acumular las calificaciones por materia y calcular el promedio. Al final, retorna un arreglo asociativo con las materias como clave y el promedio como valor.

```
<?php
class Modelo {
    private $proximas_clases = [
        ['Matemáticas', 'Lunes, 8:00 AM'],
        ['Lengua', 'Lunes, 9:00 AM'],
        ['Inglés', 'Martes, 9:00 AM'],
        ['Ciencias Sociales', 'Miércoles, 10:00 AM'],
        ['Ciencias Naturales', 'Jueves, 9:00 AM'],
        ['Arte', 'Viernes, 11:00 AM']
    ];

    private $notas_recientes = [
        ['Matemáticas', '18/20'],
        ['Lengua', '15/20'],
        ['Historia', '17/20'],
        ['Ciencias Sociales', '16/20'],
        ['Ciencias Naturales', '19/20'],
        ['Arte', '14/20']
    ];

    private $tarefas_pendientes = [
        ['Matemáticas', 'Resolver ejercicios página 45'],
        ['Lengua', 'Leer capítulo 4 del libro'],
        ['Ciencias Sociales', 'Hacer resumen del tema 2'],
        ['Ciencias Naturales', 'Investigar sobre ecosistemas'],
        ['Inglés', 'Preparar presentación oral'],
        ['Arte', 'Crear una composición artística']
    ];

    public function obtenerProximasClases() {
        return $this->proximas_clases;
    }

    public function obtenerNotasRecientes() {
        return $this->notas_recientes;
    }
}
```

```
public function obtenerTareasPendientes() {
    return $this->tareas_pendientes;
}

public function calcularPromedioNotas($notas) {
    $total_notas = count($notas);
    if ($total_notas === 0) {
        return 0;
    }

    $suma_notas = 0;
    foreach ($notas as $nota) {
        $suma_notas += (int) explode('/', $nota[1])[0];
    }

    return $suma_notas / $total_notas;
}

public function buscarClasePorDia($dia) {
    $clases_en_dia = [];
    foreach ($this->proximas_clases as $clase) {
        if (strpos(strtolower($clase[1]), strtolower($dia)) !== false) {
            $clases_en_dia[] = $clase;
        }
    }
    return $clases_en_dia;
}

public function obtenerPromediosPorMateria() {
    $promedios = [];
    foreach ($this->notas_recientes as $nota) {
        $materia = $nota[0];
        $calificacion = (int) explode('/', $nota[1])[0];
    }
}
```

```

        return $suma_notas / $total_notas;
    }

    public function buscarClasePorDia($dia) {
        $clases_en_dia = [];
        foreach ($this->proximas_clases as $clase) {
            if (strpos(strtolower($clase[1]), strtolower($dia)) !== false) {
                $clases_en_dia[] = $clase;
            }
        }
        return $clases_en_dia;
    }

    public function obtenerPromediosPorMateria() {
        $promedios = [];
        foreach ($this->notas_recientes as $nota) {
            $materia = $nota[0];
            $calificacion = (int) explode('/', $nota[1])[0];

            if (isset($promedios[$materia])) {
                $promedios[$materia]['total_notas']++;
                $promedios[$materia]['suma_notas'] += $calificacion;
            } else {
                $promedios[$materia] = [
                    'total_notas' => 1,
                    'suma_notas' => $calificacion
                ];
            }
        }

        foreach ($promedios as $materia => &$datos) {
            $datos['promedio'] = $datos['suma_notas'] / $datos['total_notas'];
        }

        return $promedios;
    }
}

```

CONTROLADOR:

Para el uso del controlador hemos aplicado estructuras condicionales para mostrar mensajes en caso de que haya o no haya tareas pendientes para el estudiante.

```
require_once 'model.php';

class Controlador {
    private $modelo;

    public function __construct() {
        $this->modelo = new Modelo();
    }

    public function obtenerDatosDashboard($tipoUsuario) {
        // Estructura condicional para diferentes tipos de usuarios
        if ($tipoUsuario == 'estudiante') {
            $proximas_clases = $this->modelo->obtenerProximasClases();
            $notas_recientes = $this->modelo->obtenerNotasRecientes();
            $tareas_pendientes = $this->modelo->obtenerTareasPendientes();
            $nota_promedio = $this->modelo->calcularPromedioNotas($notas_recientes);
        } elseif ($tipoUsuario == 'docente') {
            $proximas_clases = $this->modelo->obtenerClasesDocente();
            $notas_recientes = $this->modelo->obtenerEvaluacionesRecientes();
            $tareas_pendientes = $this->modelo->obtenerTareasPorRevisar();
            $nota_promedio = null; // No aplicable para docentes
        } else {
            $proximas_clases = [];
            $notas_recientes = [];
            $tareas_pendientes = [];
            $nota_promedio = null;
        }

        return [
            'proximas_clases' => $proximas_clases,
            'notas_recientes' => $notas_recientes,
            'tareas_pendientes' => $tareas_pendientes,
            'nota_promedio' => $nota_promedio
        ];
    }
}
```


VISTA:

Para las vistas hemos utilizado estructuras repetitivas como el `foreach`, el `while` y `do while`. Para nuestros modulos hemos considerado el uso del `while` para que nos muestre información de una base de datos mientras haya registros disponibles. En el caso del `foreach` se utiliza para interactuar con los Arrays de `datos['proximas_clases']`, `$datos['notas_recientes']` y `$datos['tareas_pendientes']` para poder mostrarlo como parte del dashboard en nuestra pagina.

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Mi Página de Estudiante</title>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
  <link href="https://cdn.jsdelivr.net/npm/@fortawesome/fontawesome-free@6.0.0-beta3/css/all.min.css" rel="stylesheet">
  <style>
    /* Estilos CSS */
    /* ... (los estilos permanecen sin cambios) ... */
  </style>
</head>
<body>
  <div class="header-container">
    <div class="menu-toggle-container">
      <button id="menu-toggle" class="menu-toggle"></button>
    </div>
    <h1>Mi Página de Estudiante</h1>
  </div>

  <div class="sidebar" id="sidebar">
    <div class="sidebar-header">MENU</div>
    <ul>
      <li><a href="#principal"><i class="fas fa-home"></i> Principal</a></li>
      <li><a href="#cursos"><i class="fas fa-book"></i> Cursos</a></li>
      <li><a href="#notas"><i class="fas fa-clipboard-check"></i> Notas</a></li>
      <li><a href="#tareas"><i class="fas fa-tasks"></i> Tareas</a></li>
      <li><a href="#horario"><i class="fas fa-clock"></i> Horario</a></li>
    </ul>
  </div>
```

```

<main>
<section id="principal">
<h2>Resumen General</h2>
<div class="resumen-dashboard">
<?php
require_once 'controller.php';

// Mostrar próximas clases
echo "<div class='dashboard-item' style='background-color: #FFDC00;'>"; // Amarillo
echo "<h3>Próximas Clases</h3>";
echo "<ul>";
foreach ($datos['proximas_clases'] as $clase) {
    echo "<li>{$clase[0]} - {$clase[1]}</li>";
}
echo "</ul>";
echo "</div>";

// Mostrar notas recientes
echo "<div class='dashboard-item' style='background-color: #FF851B;'>"; // Naranja
echo "<h3>Notas Recientes</h3>";
echo "<ul>";
foreach ($datos['notas_recientes'] as $nota) {
    echo "<li>{$nota[0]}: {$nota[1]}</li>";
}
echo "</ul>";
echo "</div>";

// Mostrar tareas pendientes
echo "<div class='dashboard-item' style='background-color: #7FDBFF;'>"; // Azul claro
echo "<h3>Tareas Pendientes</h3>";
echo "<ul>";
foreach ($datos['tareas_pendientes'] as $tarea) {
    echo "<li>{$tarea[0]}: {$tarea[1]}</li>";
}
echo "</ul>";
echo "</div>";

```

```

// Estructuras condicionales para el número de tareas
$numero_tareas = count($datos['tareas_pendientes']);
if ($numero_tareas > 5) {
    echo "<div class='alert alert-warning'>Tienes muchas tareas pendientes. ¡Es hora de empezar a trabajar!</div>";
} elseif ($numero_tareas > 0) {
    echo "<div class='alert alert-info'>Tienes algunas tareas pendientes. ¡Sigue así!</div>";
} else {
    echo "<div class='alert alert-success'>No tienes tareas pendientes. ¡Bien hecho!</div>";
}

// Mostrar promedio de notas
echo "<div class='dashboard-item' style='background-color: #2ECC40;'>"; // Verde
echo "<h3>Promedio de Notas</h3>";
if ($datos['nota_promedio'] >= 18) {
    echo "<p>Excelente trabajo. Tu promedio es {$datos['nota_promedio']}</p>";
} elseif ($datos['nota_promedio'] >= 14) {
    echo "<p>Buen trabajo. Tu promedio es {$datos['nota_promedio']}. Puedes mejorar.</p>";
} else {
    echo "<p>Necesitas mejorar. Tu promedio es {$datos['nota_promedio']}. ¡No te rindas!</p>";
}
echo "</div>";

// Uso de while y do...while
$i = 0;
echo "<div class='dashboard-item'>";
echo "<h3>Clases Repetitivas</h3>";
echo "<ul>";
while ($i < count($datos['proximas_clases'])) {
    echo "<li>{$datos['proximas_clases'][$i][0]} - {$datos['proximas_clases'][$i][1]}</li>";
    $i++;
}
echo "</ul>";
echo "</div>";

```

```

        echo "<ul>";
        do {
            echo "<li>{$datos['notas_recientes'][$j][0]}: {$datos['notas_recientes'][$j][1]}</li>";
            $j++;
        } while ($j < count($datos['notas_recientes']));
        echo "</ul>";
        echo "</div>";
    }
}

</div>
</section>
<section id="cursos" class="hidden">
    <h2>Cursos</h2>
    <p>Bienvenido a la sección de cursos.</p>
    <table>
        <tr>
            <th>Materia</th>
            <th>Docente</th>
        </tr>
        <tr>
            <td>Matemáticas</td>
            <td>Profesor García</td>
        </tr>
        <tr>
            <td>Lengua</td>
            <td>Profesor López</td>
        </tr>
        <tr>
            <td>Ciencias Sociales</td>
            <td>Profesor Martínez</td>
        </tr>
        <tr>
            <td>Ciencias Naturales</td>
            <td>Profesor Rodríguez</td>
        </tr>
        <tr>
            <td>Historia</td>
            <td>Profesor Pérez</td>
        </tr>
    </table>
</section>
<section id="notas" class="hidden">
    <h2>Notas</h2>
    <p>Bienvenido a la sección de notas.</p>
</section>
<section id="tareas" class="hidden">
    <h2>Tareas</h2>
    <p>Bienvenido a la sección de tareas.</p>
</section>
<section id="horario" class="hidden">
    <h2>Horario</h2>
    <p>Bienvenido a la sección de horario.</p>
</section>
</main>

```

Al igual que en nuestro módulo de Profesores estamos utilizando estructuras condicionales con arreglos y utilizando foreach como estructura repetitiva para que nos pueda mostrar la lista de estudiantes que tenga inscrito el profesor:

```

<div class="student-list">
  <table class="table table-bordered">
    <thead>
      <tr>
        <th>Nombre del Estudiante</th>
        <th>Matemáticas</th>
        <th>Ciencias</th>
        <th>Historia</th>
      </tr>
    </thead>
    <tbody>
      <?php
        // Mostrar la lista de estudiantes
        foreach ($SESSION['students'] as $student) {
          echo "<tr>";
          echo "<td>" . $student[0] . "</td>";
          for ($i = 1; $i < count($student); $i++) {
            echo "<td>" . $student[$i] . "</td>";
          }
          echo "</tr>";
        }
      <?>
    </tbody>
  </table>
</div>

```

```

<div class="add-student-form">
  <h3>Agregar Registro de Asistencia</h3>
  <form method="post" action="">
    <div class="form-group">
      <label for="student_name">Nombre del Estudiante</label>
      <input type="text" class="form-control" id="student_name" name="student_name" required>
    </div>
    <div class="form-group">
      <label for="math_status">Puntualidad de Matemáticas</label>
      <select class="form-control" id="math_status" name="math_status" required>
        <option value="Presente">Presente</option>
        <option value="Ausente">Ausente</option>
      </select>
    </div>
  </form>
</div>

```

```

[["Sofía Martínez", 18, 20, 14],
 ["Juan Rodríguez", 15, 17, 16],
 ["Ana Pérez", 16, 19, 13],
 ["Carlos López", 19, 18, 15],
 ["María García", 17, 16, 18],
 ["Pablo Fernández", 20, 19, 17],
 ["Laura González", 16, 15, 14],
 ["David Sánchez", 18, 17, 19],
 ["María López", 17, 18, 16],
 ["Lucía Martínez", 19, 20, 18],
 ["Maykol", 19, 20, 18]
],
];

if ($_SERVER["REQUEST_METHOD"] == "POST") {
  $studentName = $_POST["student_name"];
  $mathGrade = $_POST["math_grade"];
  $scienceGrade = $_POST["science_grade"];
  $historyGrade = $_POST["history_grade"];

  // Agregar el nuevo estudiante a la lista
  $newStudent = [$studentName, $mathGrade, $scienceGrade, $historyGrade];
  array_push($students, $newStudent);
}

foreach ($students as $index => $student) {
  echo "<tr>";
  echo "<th>" . $student[0] . "</th>";
  for ($i = 1; $i < count($student); $i++) {
    echo "<td>" . $student[$i] . "</td>";
  }
  echo "</tr>";
}
<?>
</tbody>
</table>

<form method="post" action="">
  <div class="mb-3">
    <label for="student_name" class="form-label">Nombre del Estudiante</label>
    <input type="text" class="form-control" id="student_name" name="student_name" required>
  </div>
  <div class="mb-3">
    <label for="math_grade" class="form-label">Nota de Matemáticas</label>
    <input type="number" class="form-control" id="math_grade" name="math_grade" min="0" max="20" required>
  </div>
  <div class="mb-3">
    <label for="science_grade" class="form-label">Nota de Ciencias</label>
    <input type="number" class="form-control" id="science_grade" name="science_grade" min="0" max="20" required>
  </div>
</form>

```

Recursos adicionales:

Para este proyecto estamos utilizando una instancia de Amazon Web Service (AWS) para poder desplegar nuestra página de manera pública en internet. Para poder tener actualizando nuestro sistema, estamos haciendo uso de github actions, para poder realizar el despliegue continuo en caso de las actualizaciones en nuestro código, hacia nuestro servidor. Esto nos permite actualizar rápidamente nuestra página sin tener que actualizar manualmente los archivos del proyecto que se encuentran en la nube bajo las buenas prácticas.

GLOSARIO:

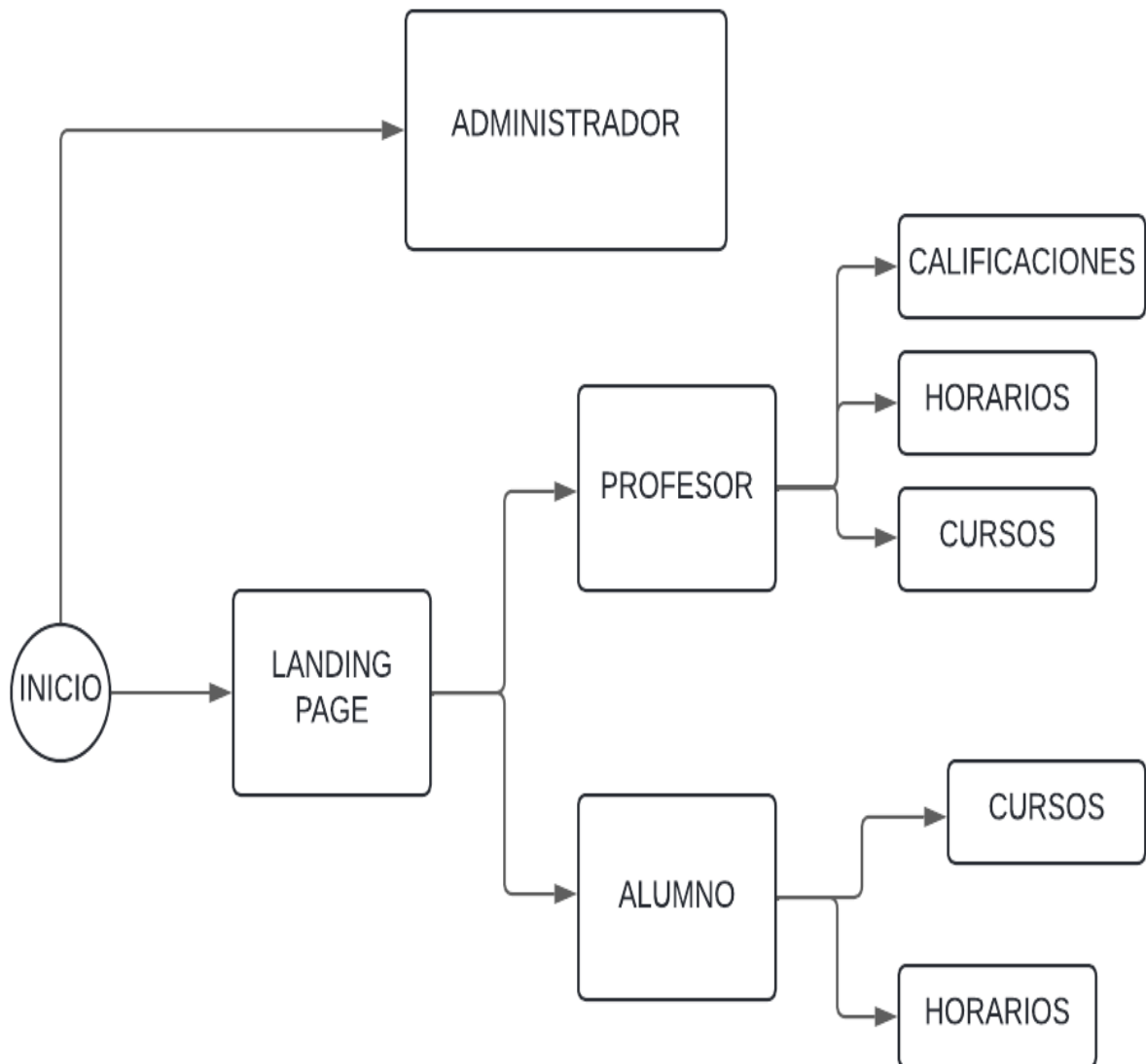
- **Dominio:** Es el nombre único que identifica a un sitio web en Internet. Es la dirección que los usuarios escriben en su navegador para acceder a un sitio web específico. Un dominio puede terminar en varias extensiones, como .com, .org, .edu, y en este caso .pe para identificar el sitio web como peruano.
- **Hosting:** Es el servicio que permite alojar y almacenar todos los archivos y datos necesarios para que un sitio web funcione y sea accesible en Internet. Un buen servicio de hosting garantiza la disponibilidad, seguridad y rapidez de acceso a la web.
- **Landing Page:** Es una página web diseñada específicamente para recibir y captar la atención de los visitantes. Su objetivo principal es convertir visitantes en leads o clientes mediante formularios de contacto, descargas, promociones, entre otros.
- **Login:** Es el proceso mediante el cual un usuario ingresa a un sistema o plataforma utilizando credenciales de acceso, como un nombre de usuario y una contraseña, para verificar su identidad y obtener acceso a áreas restringidas o personalizadas del sitio.
- **Backend:** Es la parte del desarrollo web que se encarga del manejo y procesamiento de datos en el servidor. Incluye bases de datos, servidores y aplicaciones que realizan la lógica de negocios y el almacenamiento de información. Se utiliza para gestionar la interacción entre la base de datos y la interfaz de usuario.
- **Frontend:** Es la parte del desarrollo web que se encarga de la interfaz de usuario y la experiencia del usuario en el navegador. Utiliza tecnologías como HTML, CSS y JavaScript para crear y diseñar la apariencia y comportamiento interactivo de una página web.
- **Sistema de Notas:** Es una funcionalidad específica dentro de un portal educativo que permite a los profesores registrar y gestionar las

calificaciones de los estudiantes. Los estudiantes pueden acceder a sus notas y los padres pueden hacer seguimiento del rendimiento académico.

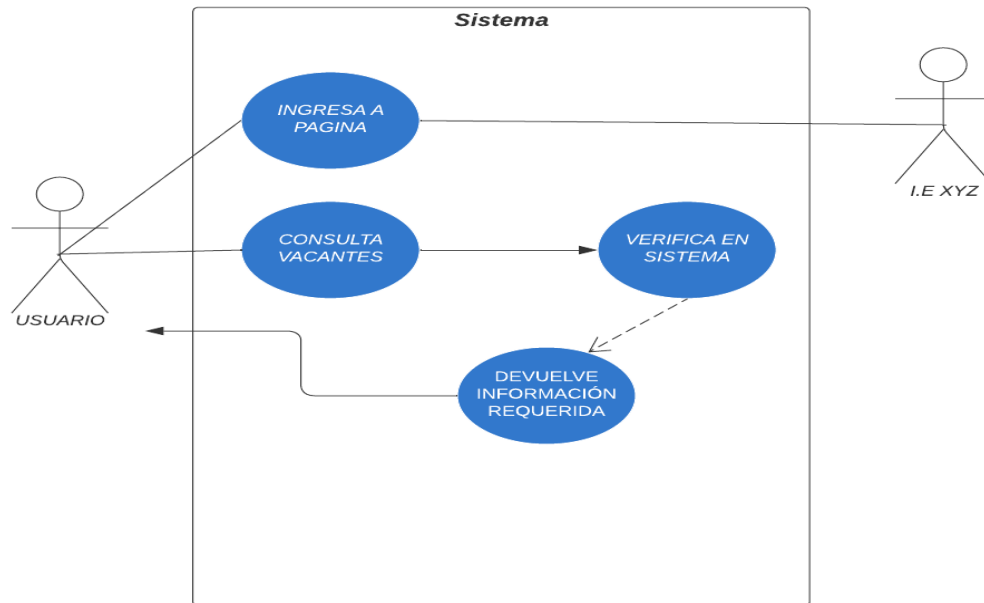
- **Arreglos:** Estructuras de datos que permiten almacenar múltiples valores en una sola variable. Los arreglos se utilizan comúnmente para manejar y manipular conjuntos de datos de manera eficiente.
- **Estructuras Repetitivas:** Son construcciones de programación que permiten ejecutar un bloque de código varias veces. Los bucles más comunes son "for", "while" y "do while", utilizados para iterar a través de arreglos o ejecutar acciones repetitivas hasta que se cumpla una condición específica.

Diagramas varios (que se han utilizado en el desarrollo del proyecto):

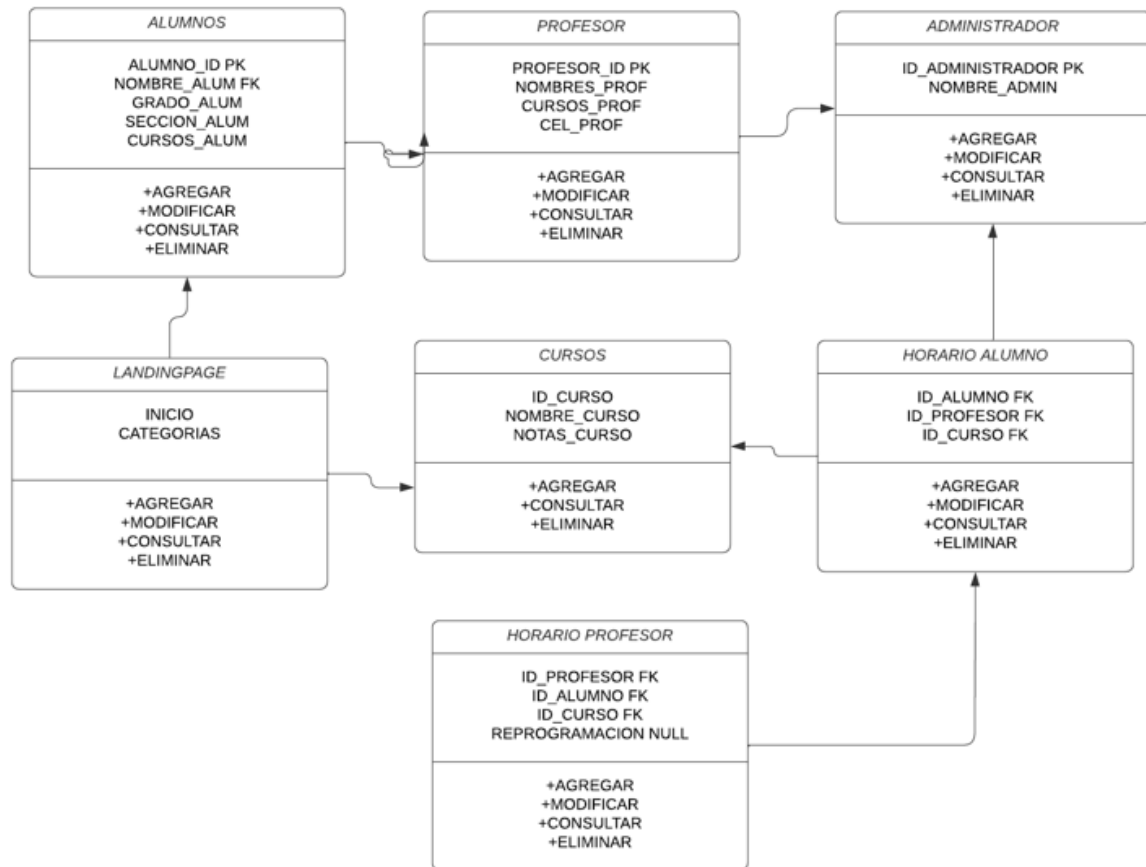
1. Esquema del Sitio Web.



2. Diagrama caso de uso



3. Modelo entidad relación de la BD



4. Sketch / Prototipos o similares.

COLEGIO XYZ



USUARIO


CONTRASEÑA

INICIAR SESION

TODOS LOS DERECHOS RESERVADOS




INICIOALUMNOSPROFESORESGRADOSCURSOS

PANEL ADMINISTRADOR

Bienvenido

PERFILCURSOSHORARIOSPROFESORES

Bienvenido

PANEL ALUMNO

246 x 232

MENU

INICIO

CURSO

CALIFICACIONES

ASISTENCIAS

~~X~~ 569

Bienvenido PROFESOR

--

ULTIMAS NOTICIAS

--