## **📄 Documentación Inicial – App Web de Control de Asistencia para "Punto 29"**

### **🏢 Información General**

* **Empresa:** Punto 29 (Restaurante)
* **Locales:** 2
* **Áreas:** Cocina, Sala, Almacén (futuro: Administración)
* **Turnos:**
  + **Mañana:** 7:30 AM – 4:30 PM (máximo hasta 5:00 PM)
  + **Noche:** 4:30 PM – 11:40 PM (máximo hasta 12:30 AM)

### **🎯 Objetivo del Proyecto**

Desarrollar una **app web dinámica** que permita:

* Registrar asistencia mediante **escaneo de QR**
* Calcular automáticamente **horas trabajadas y pagos**
* Mostrar **reportes de asistencia** por día, semana, quincena y mes
* Generar **descuentos por faltas o tardanzas**
* Premiar asistencia perfecta con **S/ 50 adicionales**
* Visualizar el historial de asistencia en un **calendario**
* Administrar perfiles de trabajadores y editar registros desde un **panel de administración**

### **🧱 Arquitectura Tecnológica**

|  |  |
| --- | --- |
| **Componente** | **Tecnología** |
| Frontend | React.js |
| Backend | Node.js + Express |
| Base de Datos | Azure SQL Database |
| Despliegue | Azure App Service |
| Repositorio | GitHub |
| Autenticación | (Por definir: JWT, Auth0, etc.) |
| Escaneo QR | Librería JS + API backend |

### **👥 Roles del Sistema**

#### **👤 Trabajador**

* Escanea QR para registrar entrada/salida
* Visualiza su perfil, historial, calendario
* Consulta su salario, descuentos, bonificaciones

#### **🛠️ Administrador**

* Edita registros de asistencia
* Crea/modifica perfiles
* Visualiza reportes generales
* Gestiona advertencias y descuentos

### **📊 Lógica de Asistencia y Pagos**

#### **✅ Registro**

* QR escaneado al ingresar y salir
* Se registra hora exacta

#### **⏱️ Cálculo de horas**

* Se calcula tiempo trabajado por día
* Se acumula por semana, quincena, mes

## **⚠️ Lógica de Tardanzas – App de Asistencia "Punto 29"**

### **⏰ Tolerancia de ingreso**

* Cada trabajador tiene **hasta 15 minutos de tolerancia** para llegar después de la hora de inicio de su turno.
  + **Turno mañana:** 7:30 AM → tolerancia hasta 7:45 AM
  + **Turno noche:** 4:30 PM → tolerancia hasta 4:45 PM

### **🟡 Tardanza leve (entre 10 y 15 minutos)**

* Si el trabajador llega **más de 10 minutos tarde** pero **dentro de los 15 minutos de tolerancia**, se registra como **tardanza leve**.
* Se permite **hasta 7 días** de tardanza leve por mes sin generar descuentos.
* A partir del **5to día**, se muestra una **advertencia** en su perfil.
* Si llega al **7mo día**, se activa el **modo de penalización**.

### **🔴 Tardanza grave (más de 15 minutos)**

* Si el trabajador llega **después de los 15 minutos de tolerancia**, se registra como **tardanza grave**.
* Cada día con tardanza grave genera un **descuento automático** en su pago.

### **🚨 Penalización por acumulación de tardanzas leves**

* Si el trabajador acumula **más de 7 días** de tardanza leve en el mes:
  + A partir del **8vo día**, **cada vez que llegue más de 10 minutos tarde**, se le aplica **descuento automático**, aunque esté dentro de los 15 minutos de tolerancia.
  + Se le muestra una **alerta persistente** en su perfil indicando que está en modo de penalización.

### **📱 Notificaciones al trabajador**

* **Día 5 de tardanza leve:** Se muestra una **advertencia** en su perfil.
* **Día 7:** Se activa el **modo de penalización**.
* **Día 8 en adelante:** Se aplican descuentos por cada nueva tardanza leve.

#### **❌ Faltas**

* Se registra como falta si no hay escaneo
* Se aplica descuento por día no trabajado

#### **🎁 Bonificación**

* Si cumple **todo el mes sin faltas ni tardanzas**, recibe **S/ 50 adicionales**

### **📅 Calendario del Trabajador**

* Muestra:
  + Hora de entrada/salida por día
  + Días trabajados
  + Días con tardanza
  + Días con falta

### **🔐 Seguridad**

* Cada trabajador tiene su perfil con acceso privado
* QR único por sesión
* Panel de administración protegido

### **🛠️ Próximos pasos**

1. Crear estructura base del proyecto (React + Node.js)
2. Definir modelo de datos en Azure SQL
3. Implementar escaneo de QR
4. Crear lógica de cálculo de horas y pagos
5. Desarrollar panel de administración
6. Desplegar en Azure App Service